User's Manual

RI Series

Real-Time Operating System

User's Manual: Start

Target Tool
RI78V4
RI850V4
RI850MP
RI600V4
RI600PX

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Electronics Corp. without notice. Please review the latest information published by Renesas Electronics Corp. through various means, including the Renesas Electronics Corp. website (http://www.renesas.com).

Renesas Electronics
www.renesas.com

Rev.1.05 Sep 2015

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.

   “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots etc.

   “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.

   Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(2012.4)

How to Use This Manual

Readers

This manual is intended for users who design and develop application systems using RL78 family, RH850 family RX family, V850 family, and 78K0R microcontrollers products.

Purpose

This manual is intended for users to understand the functions of real-time OS "RI Series" (RI78V4 V2.*xx.xx*, RI850V4 V2.*xx.xx*, RI600V4, RI600PX, RI850V4 V1.*xx.xx*, and RI850MP,RI78V4 V1.*xx.xx*) manufactured by Renesas Electronics, described the organization listed below.

Organization

This manual consists of the following major sections.

CHAPTER 1 GENERAL
CHAPTER 2 FUNCTIONS
APPENDIX A WINDOW REFERENCE

How to Read This Manual

It is assumed that the readers of this manual have general knowledge in the fields of electrical engineering, logic circuits, microcontrollers, C language, and assemblers.

To understand the hardware functions of the RL78 family, RH850 family, RX family, V850 family and 78K0R microcontrollers.
-> Refer to the **User's Manual** of each product.

Conventions

Data significance:	Higher digits on the left and lower digits on the right
Note:	Footnote for item marked with **Note** in the text
Caution:	Information requiring particular attention
Remark:	Supplementary information
Numeric representation:	Decimal ... XXXX
	Hexadecimal ... 0xXXXX

Prefixes indicating power of 2 (address space and memory capacity):

K (kilo)	$2^{10} = 1024$
M (mega)	$2^{20} = 1024^2$

Related Documents

The related documents indicated in this publication may include preliminary versions. However, preliminary versions are not marked as such.

Document Name		Document No.
RI Series	Start	This manual
	Message	R20UT0756E
RI850V4 V2.*xx.xx*	Coding	R20UT2889E
	Debug	R20UT2890E
	Analysis	R20UT2891E
RI600V4	Coding	R20UT0711E
	Debug	R20UT0775E
	Analysis	R20UT2185E
RI600PX	Coding	R20UT0964E
	Debug	R20UT0950E
RI78V4 V2.*xx.xx*	Coding	R20UT3375E
	Debug	R20UT3374E
	Analysis	R20UT3373E
RI850V4	Coding	R20UT0515E
	Debug	R20UT0754E
	Analysis	R20UT0517E
RI850MP	Coding	R20UT0519E
	Debug	R20UT0755E
RI78V4	Coding	R20UT0511E
	Debug	R20UT0753E
	Analysis	R20UT0513E

Caution The related documents listed above are subject to change without notice. Be sure to use the latest edition of each document when designing.

All trademarks or registered trademarks in this document are the property of their respective owners.

TABLE OF CONTENTS

CHAPTER 1 GENERAL ... 6

1.1 Outline ... 6
1.1.1 Real-time OS ... 6
1.1.2 Multi-task OS ... 6
1.2 Features ... 7
1.3 Configuration ... 9
1.3.1 RI78V4 V2.xx.xx ... 9
1.3.2 RI850V4 V2.xx.xx ... 10
1.3.3 RI600V4 ... 11
1.3.4 RI600PX ... 12
1.3.5 RI850V4 V1.xx.xx ... 13
1.3.6 RI850MP ... 14
1.3.7 RI78V4 V1.xx.xx ... 15
1.4 Folder Configuration ... 16
1.4.1 RI78V4 V2.xx.xx ... 16
1.4.2 RI850V4 V2.xx.xx ... 18
1.4.3 RI600V4 ... 20
1.4.4 RI600PX ... 21
1.4.5 RI850V4 V1.xx.xx ... 22
1.4.6 RI850MP ... 26
1.4.7 RI78V4 V1.xx.xx ... 28
1.5 Execution Environment ... 30

CHAPTER 2 FUNCTIONS ... 33

2.1 Installing Real-time OS Package ... 33
2.2 Uninstalling Real-time OS Package ... 34
2.3 Create a Project for RI Series ... 35
2.4 Convert a CubeSuite Project into a CS+ Project ... 36
2.5 Convert a HEW Project into a CS+ Project ... 36
2.6 Convert a Project “RI78V4 V1.xx.xx -> V2.xx.xx“ or “RI850V4 V1.xx.xx -> V2.xx.xx“ ... 36
2.7 Start Resource Information Tool ... 37
2.8 Start Task Analyzer Tool [RI78V4 V2.xx.xx][RI850V4 V2.xx.xx][RI600V4] ... 39
2.9 Start Program Analyzer Tool [RI850V4 V1.xx.xx][RI78V4 V1.xx.xx] ... 41

APPENDIX A WINDOW REFERENCE ... 43

A.1 Description ... 43

CHAPTER 1 GENERAL

This chapter describes the outline of real-time OS "RI series" manufactured by Renesas Electronics.

1.1 Outline

The RI series is a built-in real-time, multi-task OS that provides a highly efficient real-time, multi-task environment to increases the application range of processor control units.

The RI series is a high-speed, compact OS capable of being stored in and run from the ROM of a target system.

1.1.1 Real-time OS

Control equipment demands systems that can rapidly respond to events occurring both internal and external to the equipment. Conventional systems have utilized simple interrupt handling as a means of satisfying this demand. As control equipment has become more powerful, however, it has proved difficult for systems to satisfy these requirements by means of simple interrupt handling alone.

In other words, the task of managing the order in which internal and external events are processed has become increasingly difficult as systems have increased in complexity and programs have become larger.

Real-time OS has been designed to overcome this problem.

The main purpose of a real-time OS is to respond to internal and external events rapidly and execute programs in the optimum order.

1.1.2 Multi-task OS

A "task" is the minimum unit in which a program can be executed by an OS. "Multi-task" is the name given to the mode of operation in which a single processor processes multiple tasks concurrently.

Actually, the processor can handle no more than one program (instruction) at a time. But, by switching the processor's attention to individual tasks on a regular basis (at a certain timing) it appears that the tasks are being processed simultaneously.

A multi-task OS enables the parallel processing of tasks by switching the tasks to be executed as determined by the system.

One important purpose of a multi-task OS is to improve the throughput of the overall system through the parallel processing of multiple tasks.

1.2 Features

The RI series has the following features.

(1) Conformity with μITRON4.0 specification

The RI series is designed as a typical built-in control OS architecture that conform to the μITRON4.0 specification.

(2) High versatility

To support various execution environments, the RI series extracts hardware-dependent processing that is required to execute processing as user-own coding modules and target-dependent modules, and provides it as sample source files. This enhances portability for various execution environments and facilitates customization as well.

(3) Compact design

The RI series is a real-time, multi-task OS that has been designed on the assumption that it will be incorporated into the target system; it has been made as compact as possible to enable it to be loaded into a system's ROM. Since it is possible to link only those service calls that are used by the user within the system among the service calls provided by the RI series during system building, a real-time multitask OS that is ideally suited to the needs of the user while being compact can be built.

(4) Support tools related to real-time OS

The RI series provides tools that are useful during system building and system debugging.

(a) Configurator

Loads highly writable and readable system configuration files as input files, and outputs information files (system information table file, system information header file, etc.) as information files.

Remark See "RI78V4 V2.*xx.xx*/RI850V4 V2.*xx.xx*/RI600V4/RI600PX/RI850V4/RI850MP/RI78V4 Real-Time Operating System User's Manual: Coding" for details.

(b) Table generation utility [RI600PX]

Table generation utility provides the command line tool that by gathering the service call information used by an application, generates the service call and interrupt vector tables most suitable for the application.

Remark See "RI600PX Real-Time Operating System User's Manual: Coding" for details.

(c) Resource information tool

Resource information tool provides functions for efficient system debugging (OS resource display function, etc.).

Remark See "RI78V4 V2.*xx.xx*/RI850V4 V2.*xx.xx*/RI600V4/RI600PX/RI850V4/RI850MP/RI78V4 Real-Time Operating System User's Manual: Debug" for details.

(d) Task analyzer tool [RI78V4 V2.*xx.xx*] [RI850V4 V2.*xx.xx*] [RI600V4]

The task analyzer tool receives trace information from a debugger and provides functions for graphically displaying analysis information on the status of execution transition of programs being processed, usage of real-time OS resources, usage of the CPU, etc..

Remark See "RI78V4 V2.*xx.xx*/RI850V4 V2.*xx.xx*/RI600V4 Real-Time Operating System User's Manual: Analysis" for details.

(e) Program analyzer [RI850V4 V1.*xx.xx*] [RI78V4 V1.*xx.xx*]

System program analyzer provides functions for performing quantitative performance analysis on processing programs (analysis related to time such as bugs in processing timing or performance evaluation of entire system).

Remark See "RI850V4/RI78V4 Real-Time Operating System User's Manual: Analysis" for details.

(5) Memory protection function [RI600PX]

(a) High-reliability system

To reduce a possibility of being unable to detect program glitches when debugging the program and causing a trouble in the market after the system has been shipped from the factory, this OS assures the system of high-reliability. If memory data destruction occurs especially in a memory area in which the OS, etc. are stored, the system may produce a dangerous condition by, for example, operating erratically. However, since RI600PX is free of memory data corruptions, the system can continue operating normally, and is therefore assured of high system reliability.

(b) Debug assistance

In systems without memory protection, a corruption of memory content by an illegal pointer behavior, etc. generally is not noticed until it actually comes to the surface as a trouble symptom. The cause of a bug can only be identified by analyzing emulator's trace data, which requires a large amount of time. The RI600PX can detect a bug when an illegal memory access is committed, enabling the debugging efficiency to be greatly increased.

Remark See "RI600PX Real-Time Operating System User's Manual: Coding" for details.

(6) Support for programming for dual-core microcontrollers [RI850MP]

The RI series provides the service for achieving synchronization between processor elements (PEs) so as to control a dual-core microcontroller.

Remark See "RI850MP Real-Time Operating System User's Manual: Coding" for details.

1.3 Configuration

The module construction of the RI series is as follows.

1.3.1 RI78V4 V2.*xx.xx*

The RI78V4 V2.*xx.xx* consists of the following two types of modules.

(1) Kernel

The kernel, which is the processing block that forms the core of the RI78V4 V2.*xx.xx* and the main processing block for the service calls provided by the RI78V4, provides the following functions.

- Task management functions
- Task dependent synchronization functions
- Synchronization and communication functions (Semaphores, eventflags, data queues, mailboxes)
- Memory pool management functions (Fixed-sized memory pools)
- Time management functions
- System state management functions
- Interrupt management functions
- System configuration management functions
- Scheduler

(2) User-own coding module

To support various execution environments, the RI78V4 V2.*xx.xx* extracts hardware-dependent processing that is required to execute processing as user-own coding modules, and provides it as sample source files. This enhances portability for various execution environments and facilitates customization as well.
The following lists the user-own coding modules extracted for each function.

- Interrupt management functions (Interrupt entry processing)
- System configuration management functions (Boot processing, initialization routine)
- Scheduler (Idle routine)

Remark For interrupt handlers written by C language(defined the attribution “TA_HLNG“ in interrupt handler definition “DEF_INH” in system configuration file), the user is not required to write the relevant interrupt entry processing because the C compiler automatically outputs the interrupt entry processing corresponding to the interrupt request name.

Remark See "RI78V4 V2.*xx.xx* Real-Time Operating System User's Manual: Coding" for details.

1.3.2 RI850V4 V2.*xx.xx*

The RI850V4 V2.*xx.xx* consists of the following modules.

(1) Kernel

The kernel, which is the processing block that forms the core of the RI850V4 V2.*xx.xx* and the main processing block for the service calls provided by the RI850V4, provides the following functions.

- Task management functions
- Task dependent synchronization functions
- Synchronization and communication functions (Semaphores, eventflags, data queues, mailboxes)
- Extended synchronization and communication functions (Mutexes)
- Memory pool management functions (Fixed-sized memory pools, variable-sized memory pools)
- Time management functions
- System state management functions
- Interrupt management functions
- Service call management functions
- System configuration management functions
- Scheduler
- System initialization functions

(2) User-own coding module

To support various execution environments, the RI850V4 V2.*xx.xx* extracts hardware-dependent processing that is required to execute processing as user-own coding modules, and provides it as sample source files. This enhances portability for various execution environments and facilitates customization as well.

The following lists the user-own coding modules extracted for each function.

- Task management functions (Post-stack overflow processing)
- Interrupt management functions (Interrupt entry processing)
- System configuration management functions (CPU exception entry processing, initialization routine)
- Scheduler (Idle routine)
- System initialization routine (Boot processing)
- System initialization routine (System dependent information)

Remark See "RI850V4 V2.xx.xx Real-Time Operating System User's Manual: Coding" for details.

1.3.3 RI600V4

The RI600V4 consists of the following modules.

(1) Kernel

The kernel, which is the processing block that forms the core of the RI600V4 and the main processing block for the service calls provided by the RI600V4, provides the following functions.

- Task management functions
- Task dependent synchronization functions
- Synchronization and communication functions (Semaphores, eventflags, data queues, mailboxes)
- Extended synchronization and communication functions (Mutexes, message buffer)
- Memory pool management functions (Fixed-sized memory pools, variable-sized memory pools)
- Time management functions
- System state management functions
- Interrupt management functions
- System configuration management functions
- Object reset module
- Scheduler

(2) User-own coding module

To support various execution environments, the RI600V4 extracts hardware-dependent processing etc. that is required to execute processing as user-own coding modules, and provides it as sample source files. This enhances portability for various execution environments and facilitates customization as well.

The following lists the user-own coding modules extracted for each function.

- System down (System down routine)
- System initializing process (Boot processing, section information)

Remark See "RI600V4 Real-Time Operating System User's Manual: Coding" for details.

1.3.4 RI600PX

The RI600PX consists of the following modules.

(1) Kernel

The kernel, which is the processing block that forms the core of the RI600PX and the main processing block for the service calls provided by the RI600PX, provides the following functions.

- Task management functions
- Task dependent synchronization functions
- Task exception handling functions
- Synchronization and communication functions (Semaphores, eventflags, data queues, mailboxes)
- Extended synchronization and communication functions (Mutexes, message buffer)
- Memory pool management functions (Fixed-sized memory pools, variable-sized memory pools)
- Time management functions
- System state management functions
- Interrupt management functions
- System configuration management functions
- Object reset module
- Memory object management functions
- Scheduler

(2) User-own coding module

To support various execution environments, the RI600PX extracts hardware-dependent processing etc. that is required to execute processing as user-own coding modules, and provides it as sample source files. This enhances portability for various execution environments and facilitates customization as well.

The following lists the user-own coding modules extracted for each function.

- Memory protection functions (Access exception handler)
- Time management functions (Base clock timer initialization routine)
- System down (System down routine)
- System initializing process (Boot processing, section information)

Remark See "RI600PX Real-Time Operating System User's Manual: Coding" for details.

1.3.5 RI850V4 V1.*xx.xx*

The RI850V4 V1.*xx.xx* consists of the following modules.

(1) Kernel

The kernel, which is the processing block that forms the core of the RI850V4 V1.*xx.xx* and the main processing block for the service calls provided by the RI850V4 V1.*xx.xx*, provides the following functions.

- Task management functions
- Task dependent synchronization functions
- Task exception handling functions
- Synchronization and communication functions (Semaphores, eventflags, data queues, mailboxes)
- Extended synchronization and communication functions (Mutexes)
- Memory pool management functions (Fixed-sized memory pools, variable-sized memory pools)
- Time management functions
- System state management functions
- Interrupt management functions
- Service call management functions
- System configuration management functions
- Scheduler

(2) Target-dependent module

To support various execution environments, the RI850V4 V1.*xx.xx* extracts hardware-dependent processing that is required to execute processing as target-dependent modules, and provides them as sample source files. This enhances portability for various execution environments and facilitates customization as well.

The following lists the target-dependent modules extracted for each function.

- Interrupt management functions (Service call "dis_int", service call "ena_int", interrupt mask setting processing (overwrite setting), interrupt mask setting processing (OR setting), interrupt mask acquire processing)

(3) User-own coding module

To support various execution environments, the RI850V4 V1.*xx.xx* extracts hardware-dependent processing that is required to execute processing as user-own coding modules, and provides it as sample source files. This enhances portability for various execution environments and facilitates customization as well.

The following lists the user-own coding modules extracted for each function.

- Task management functions (Post-stack overflow processing)
- Interrupt management functions (Interrupt entry processing)
- System configuration management functions (CPU exception entry processing, initialization routine)
- Scheduler (Idle routine)
- System initialization routine (Boot processing)

Remark See "RI850V4 Real-Time Operating System User's Manual: Coding" for details.

1.3.6 RI850MP

The RI850MP consists of the following modules.

(1) Kernel

The kernel, which is the processing block that forms the core of the RI850MP and the main processing block for the service calls provided by the RI850MP, provides the following functions.

- Task management functions
- Task dependent synchronization functions
- Synchronization and communication functions (Semaphores, eventflags, data queues, mailboxes)
- Extended synchronization and communication functions (Mutexes)
- Memory pool management functions (Fixed-sized memory pools)
- Time management functions
- System state management functions
- Interrupt management functions
- System configuration management functions
- Scheduler

(2) Target-dependent module

To support various execution environments, the RI850MP extracts hardware-dependent processing that is required to execute processing as target-dependent modules, and provides them as sample source files. This enhances portability for various execution environments and facilitates customization as well.

The following lists the target-dependent modules extracted for each function.

- Interrupt management functions (Service call "dis_int", service call "ena_int", interrupt mask setting processing (overwrite setting), interrupt mask setting processing (OR setting),interrupt mask acquire processing)

(3) User-own coding module

To support various execution environments, the RI850MP extracts hardware-dependent processing that is required to execute processing as user-own coding modules, and provides it as sample source files. This enhances portability for various execution environments and facilitates customization as well.

The following lists the user-own coding modules extracted for each function.

- Interrupt management functions (Interrupt entry processing)
- System configuration management functions (CPU exception entry processing, initialization routine)
- Scheduler (Idle routine)
- System initialization routine (Boot processing)

Remark See "RI850MP Real-Time Operating System User's Manual: Coding" for details.

1.3.7 RI78V4 V1.*xx.xx*

The RI78V4 V1.*xx.xx* consists of the following modules.

(1) Kernel

The kernel, which is the processing block that forms the core of the RI78V4 V1.*xx.xx* and the main processing block for the service calls provided by the RI78V4 V1.*xx.xx*, provides the following functions.

- Task management functions
- Task dependent synchronization functions
- Synchronization and communication functions (Semaphores, eventflags, mailboxes)
- Memory pool management functions (Fixed-sized memory pools)
- Time management functions
- System state management functions
- Interrupt management functions
- System configuration management functions
- Scheduler

(2) User-own coding module

To support various execution environments, the RI78V4 extracts hardware-dependent processing that is required to execute processing as user-own coding modules, and provides it as sample source files. This enhances portability for various execution environments and facilitates customization as well.

The following lists the user-own coding modules extracted for each function.

- Interrupt management functions (Interrupt entry processing)
- System configuration management functions (Boot processing, initialization routine)
- Scheduler (Idle routine)

Remark For interrupt handlers written using the #pragma rtos_interrupt directive, the user is not required to write the relevant interrupt entry processing because the C compiler automatically outputs the interrupt entry processing corresponding to the interrupt request name.

Remark See "RI78V4 Real-Time Operating System User's Manual: Coding" for details.

1.4 Folder Configuration

This section explains the folder configuration of the files read from the supply medium when RI series has been installed. The RI series is supplied in the form of an object release version or a source release version.

- Object release version (A file of executable format of real-time OS is supplied.)
- Source release version (To generate a library of real-time OS, a source file is supplied.)

1.4.1 RI78V4 V2.*xx.xx*

The following shows the folder configuration when the files stored in the RI78V4 V2.*xx.xx* distribution media have been installed.

Figure 1-1. Folder Configuration (Object Release Version) [RI78V4 V2.*xx.xx*]

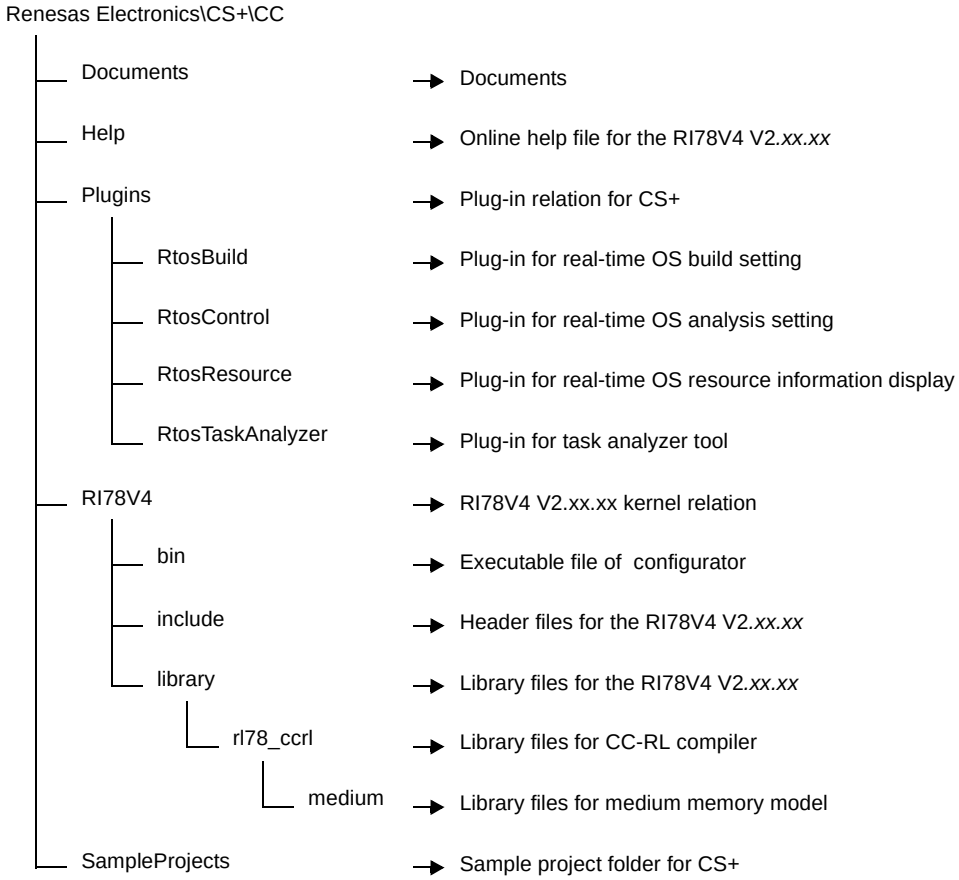
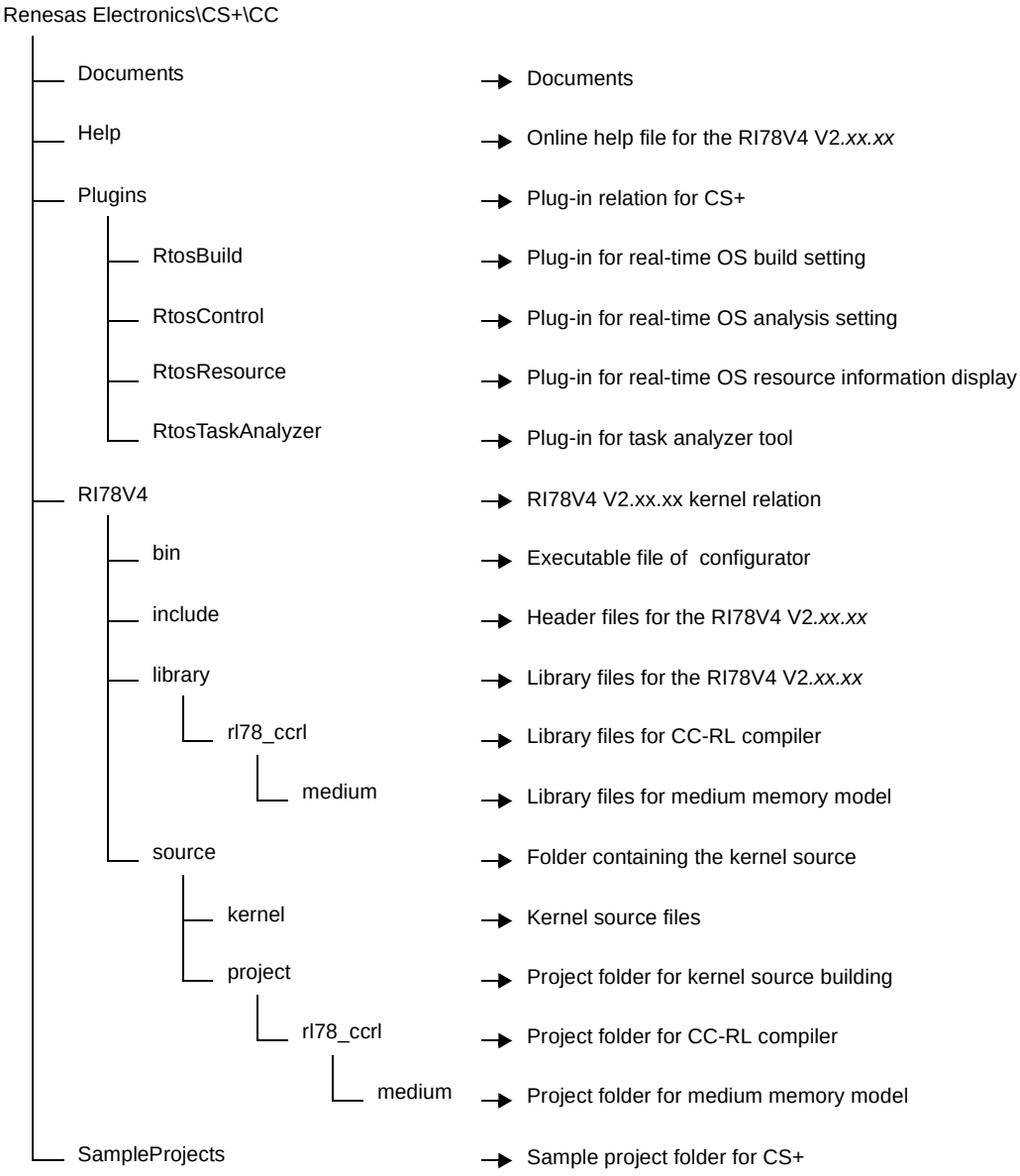


Figure 1-2. Folder Configuration (Source Release Version) [RI78V4 V2.*xx.xx*]



1.4.2 RI850V4 V2.*xx.xx*

The following shows the folder configuration when the files stored in the RI850V4 V2.*xx.xx* distribution media have been installed.

Figure 1-3. Folder Configuration (Object Release Version) [RI850V4 V2.*xx.xx*]

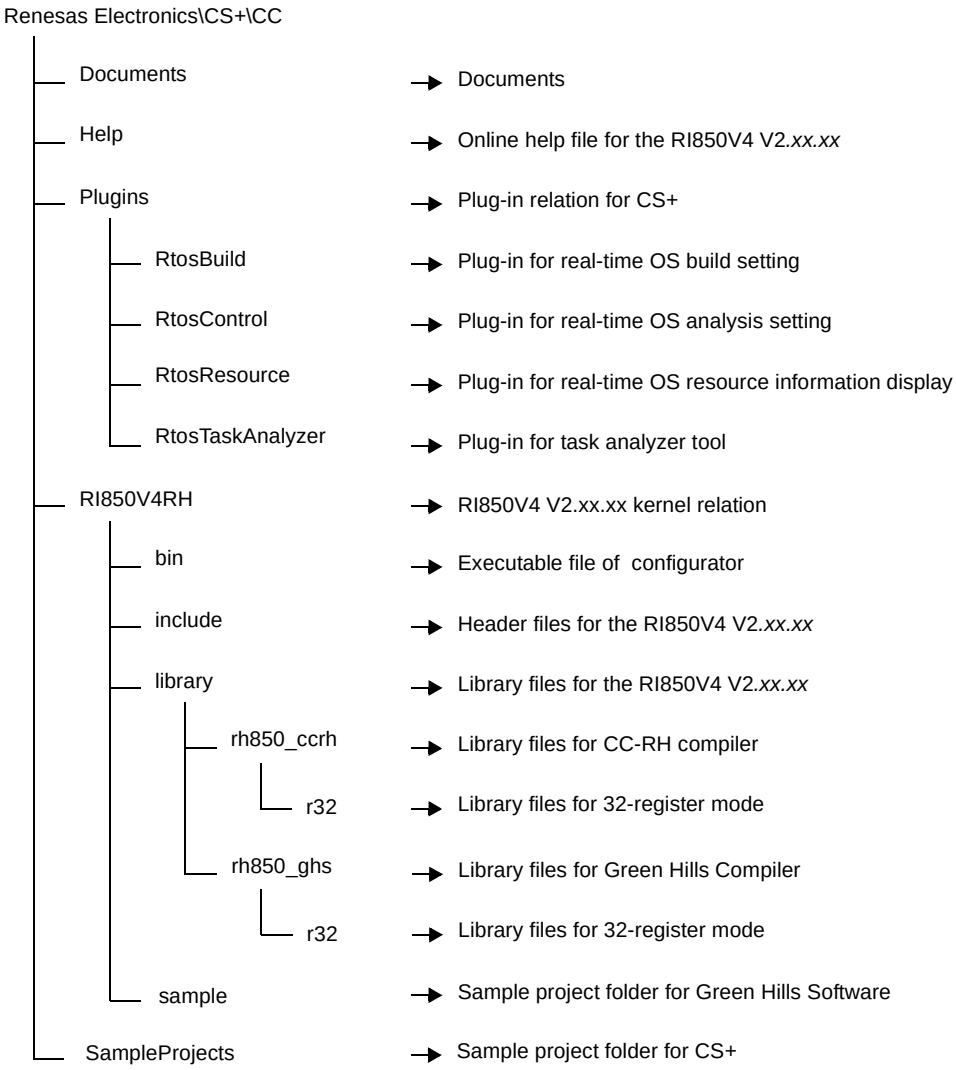
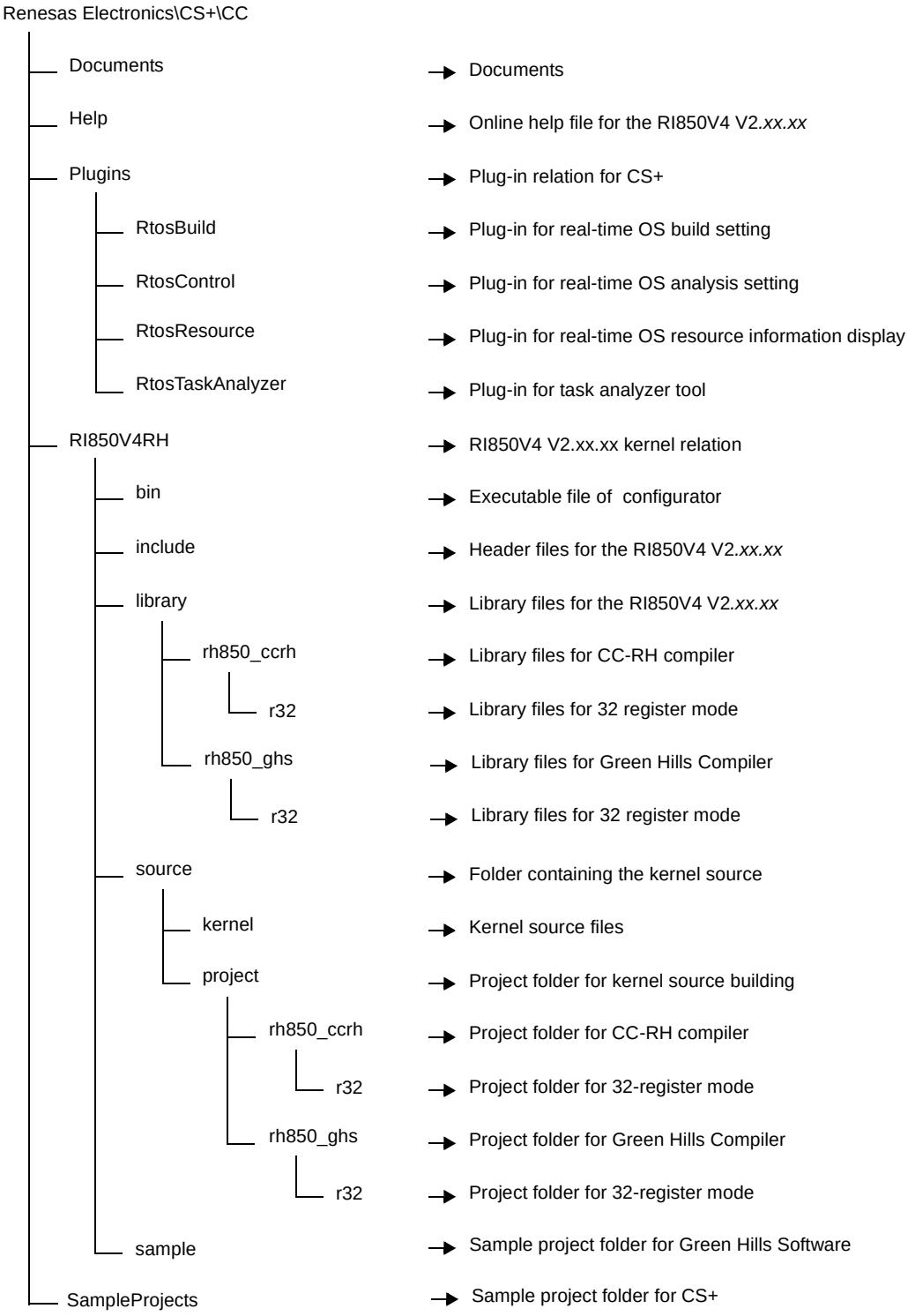


Figure 1-4. Folder Configuration (Source Release Version) [RI850V4 V2.*xx.xx*]



1.4.3 RI600V4

The following shows the folder configuration when the files stored in the RI600V4 distribution media have been installed.

Figure 1-5. Folder Configuration (Object Release Version) [RI600V4]

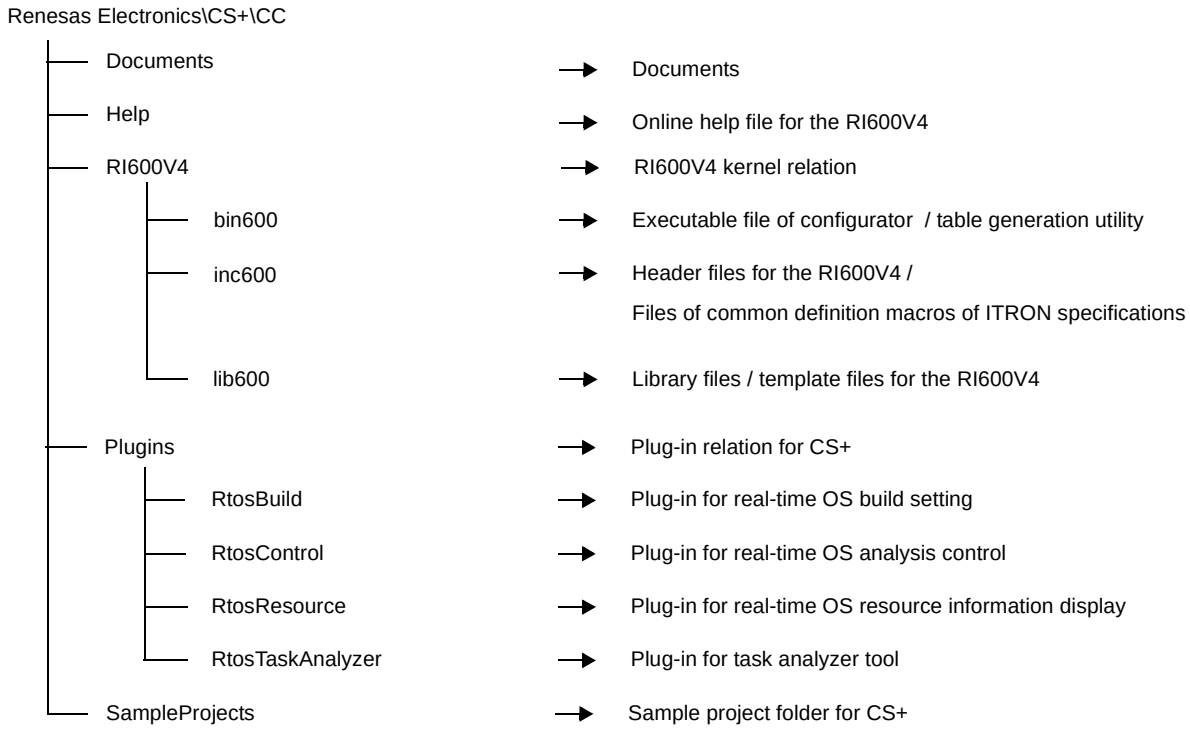
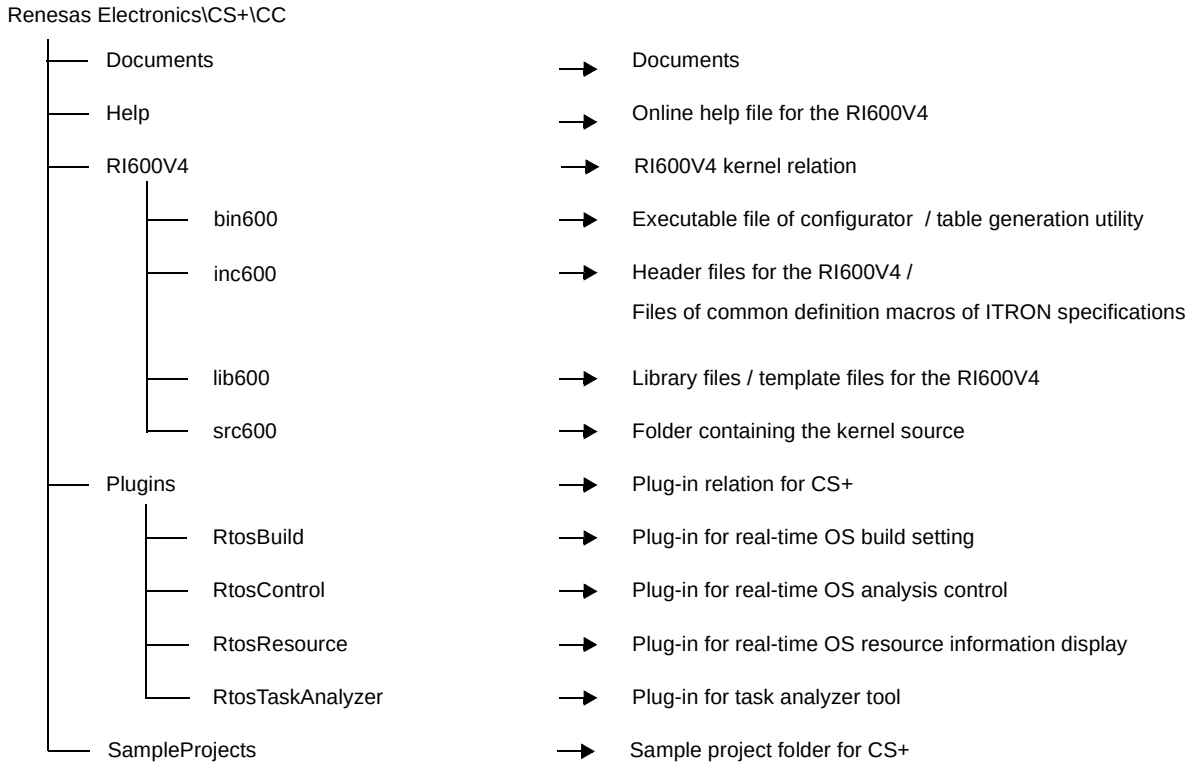


Figure 1-6. Folder Configuration (Source Release Version) [RI600V4]



1.4.4 RI600PX

The following shows the folder configuration when the files stored in the RI600PX distribution media have been installed.

Figure 1-7. Folder Configuration (Object Release Version) [RI600PX]

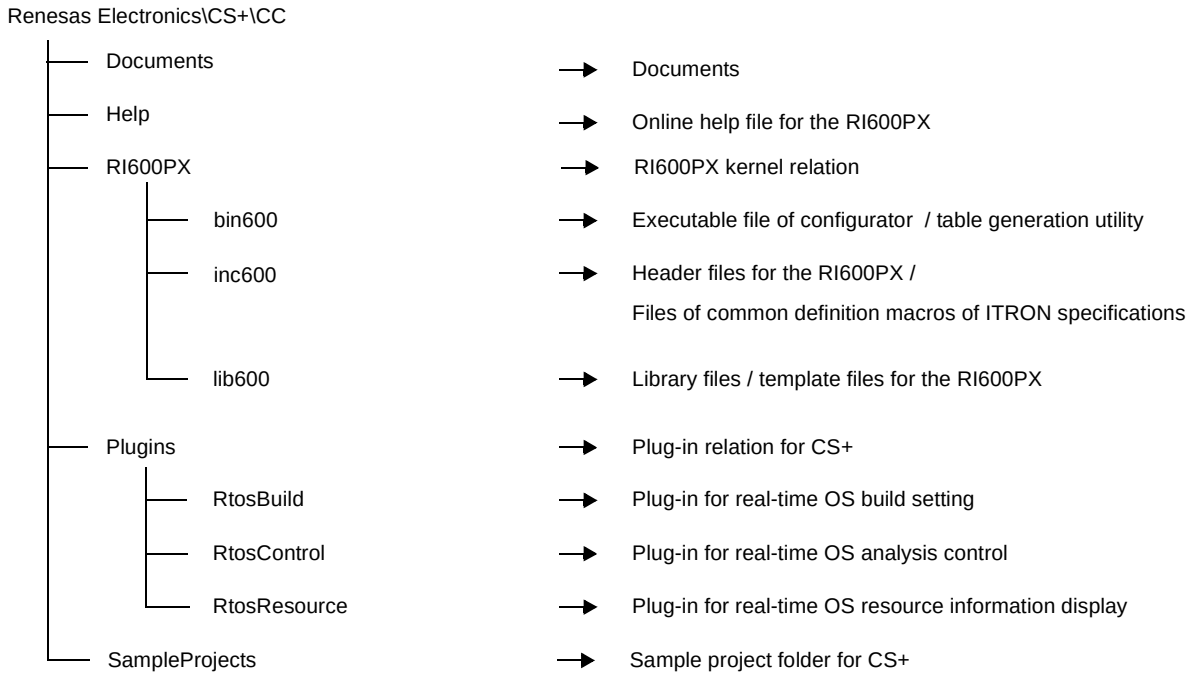
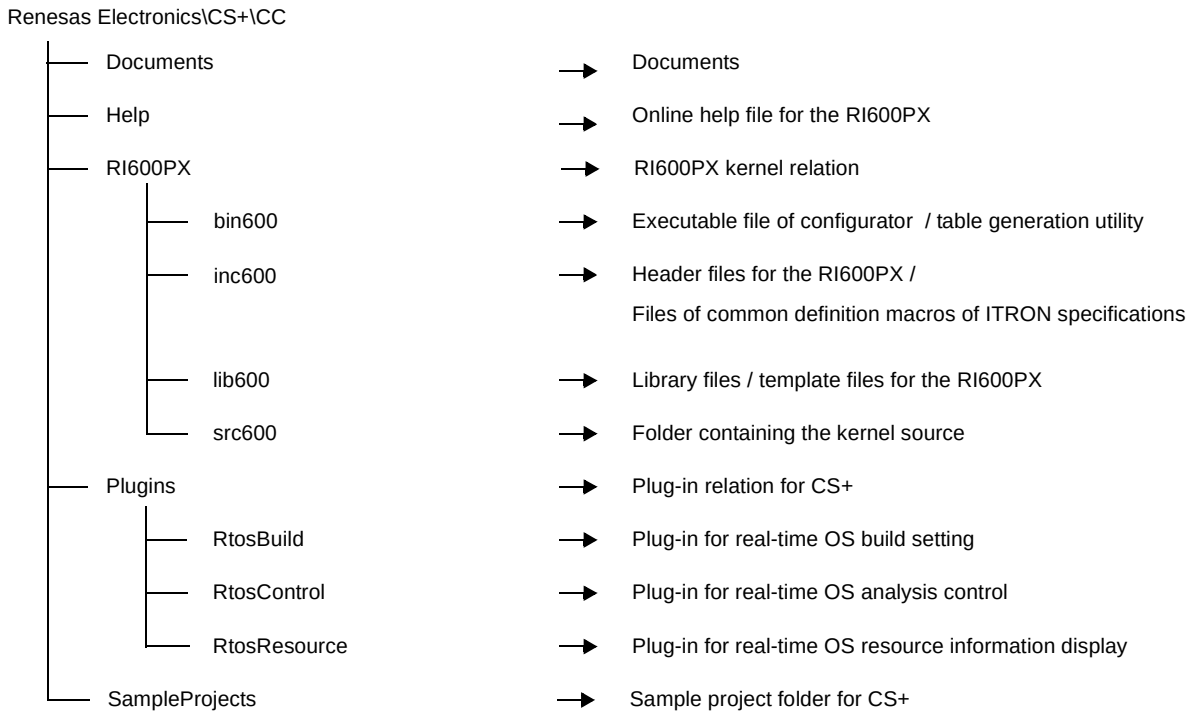


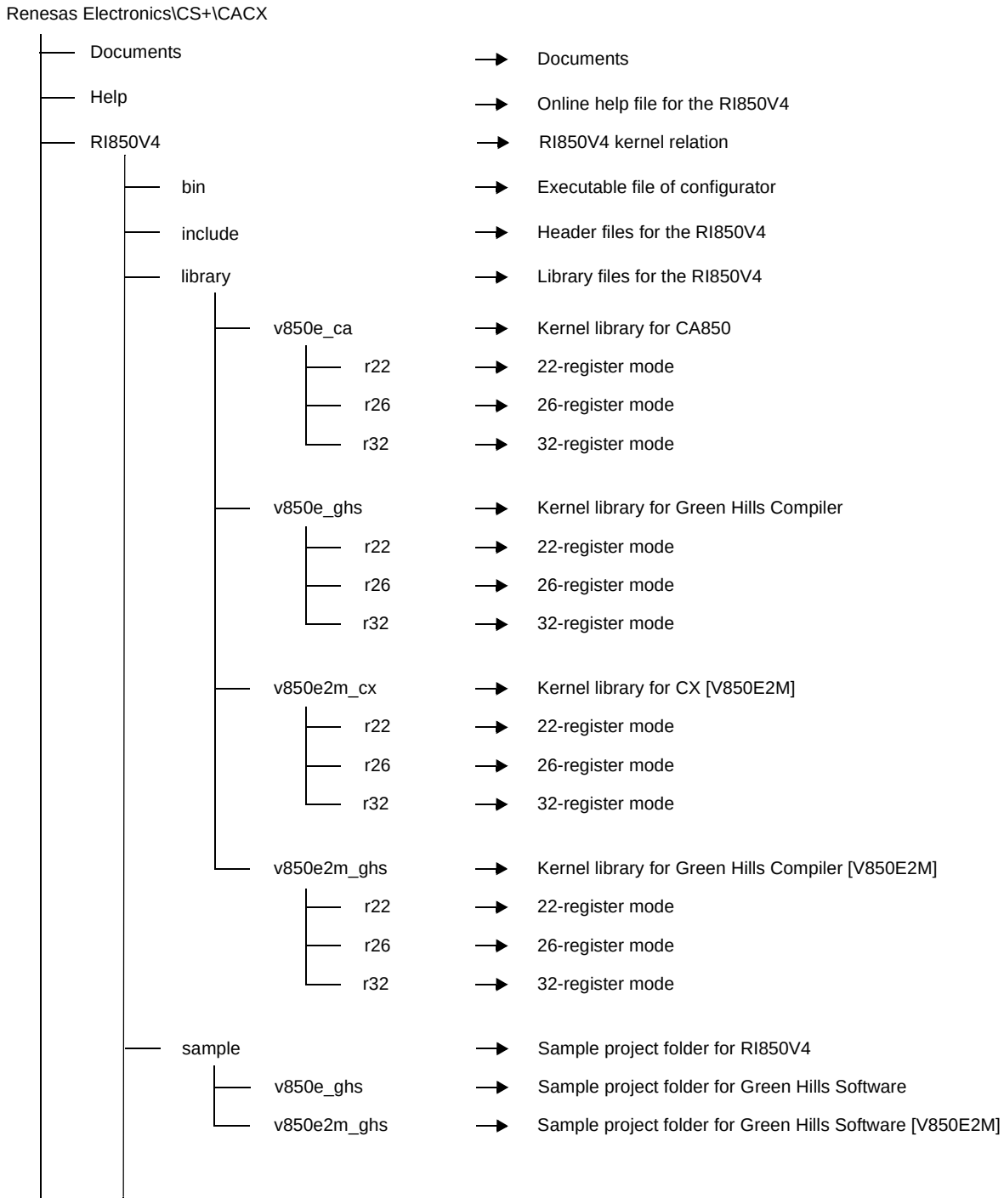
Figure 1-8. Folder Configuration (Source Release Version) [RI600PX]



1.4.5 RI850V4 V1.*xx.xx*

The following shows the folder configuration when the files stored in the RI850V4 V1.*xx.xx* distribution media have been installed.

Figure 1-9. Folder Configuration (Object Release Version) [RI850V4 V1.*xx.xx*]



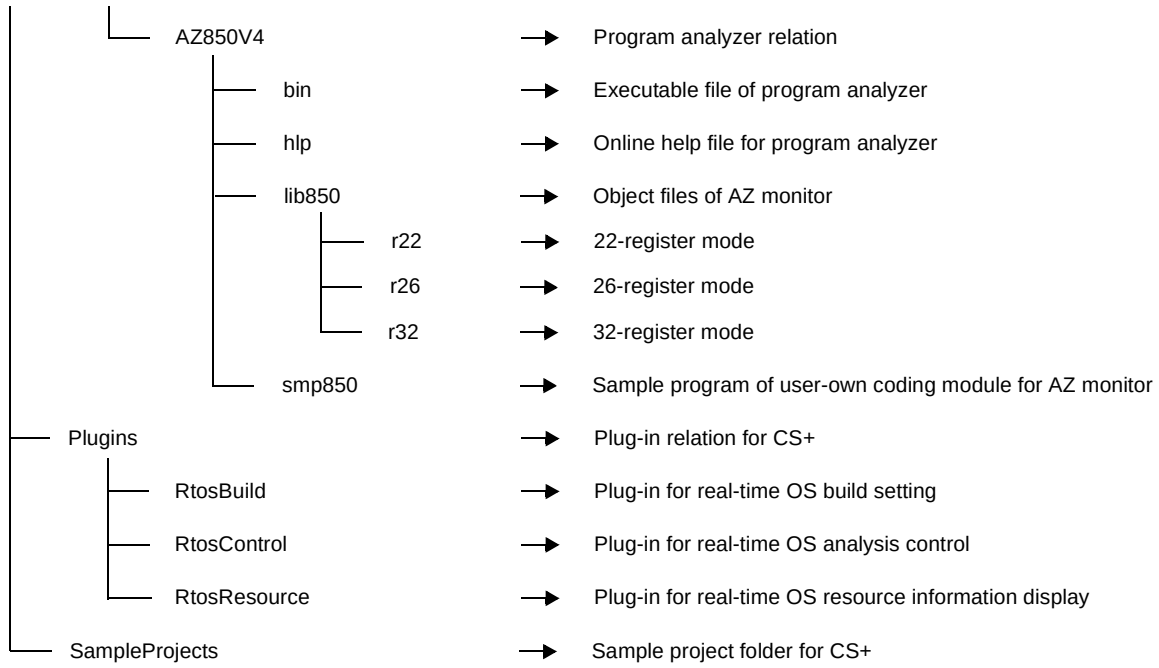
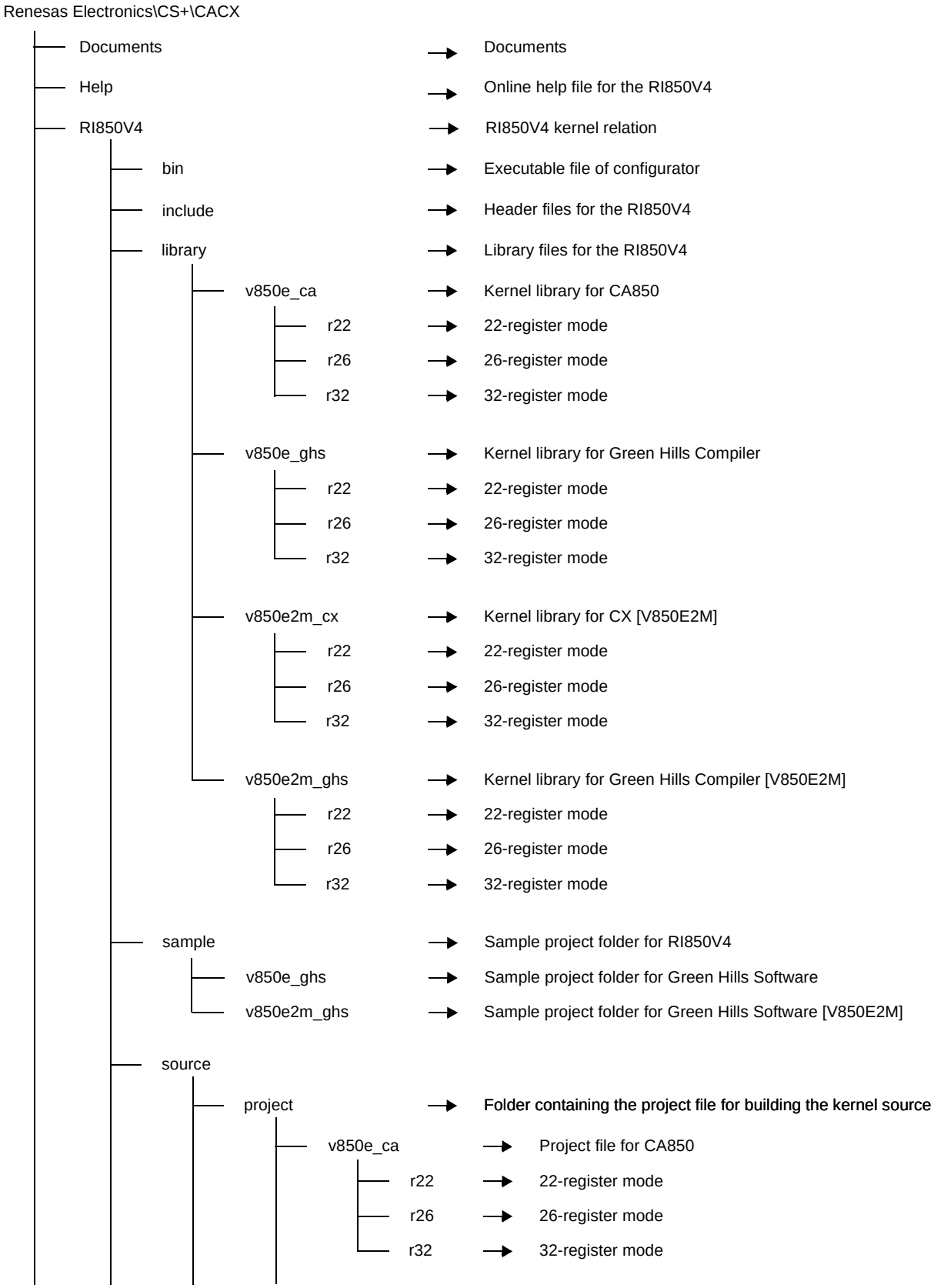
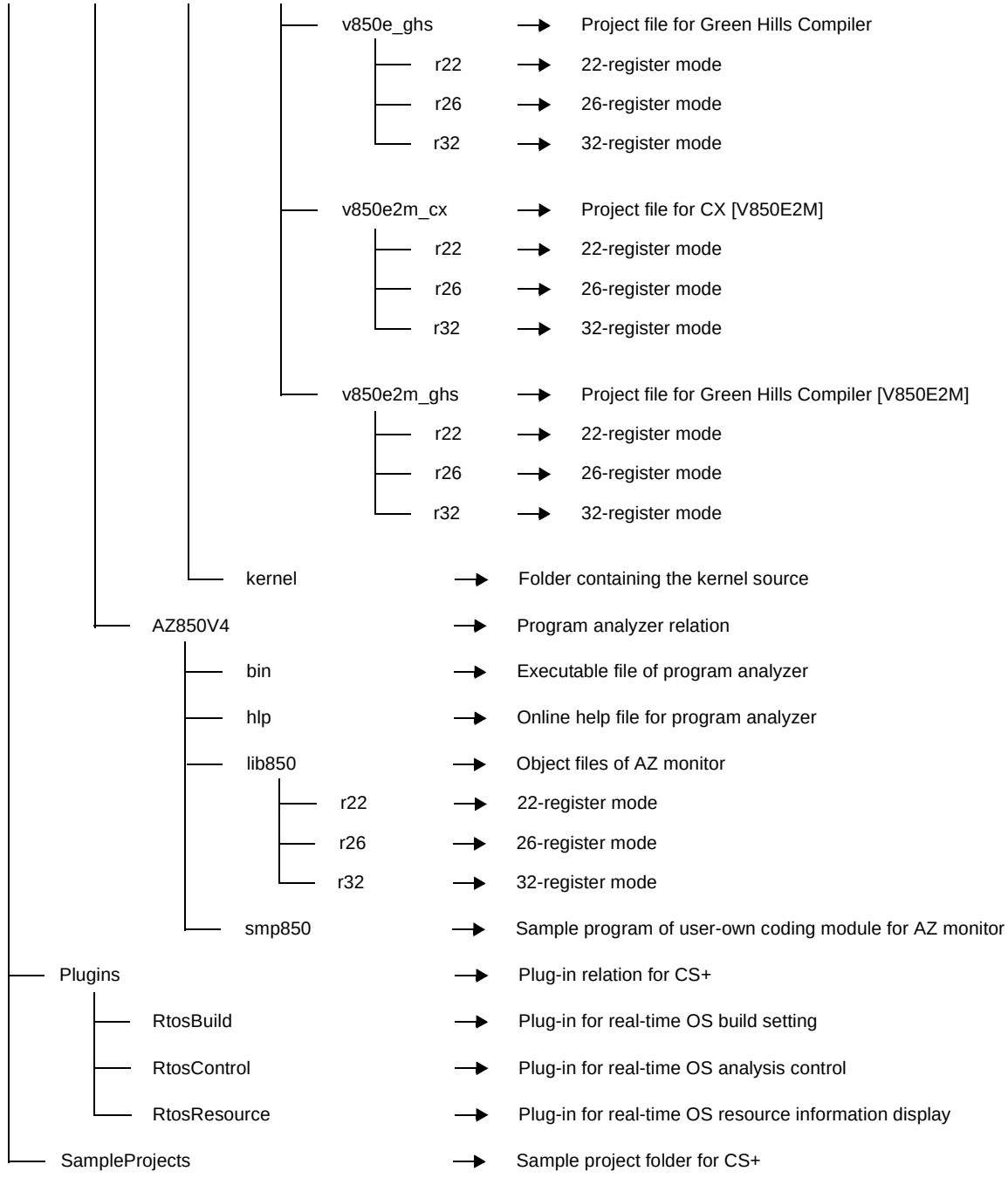

AZ850V4
Program analyzer relation
bin
Executable file of program analyzer
hlp
Online help file for program analyzer
lib850
Object files of AZ monitor
r22
22-register mode
r26
26-register mode
r32
32-register mode
smp850
Sample program of user-own coding module for AZ monitor
Plugins
Plug-in relation for CS+
RtosBuild
Plug-in for real-time OS build setting
RtosControl
Plug-in for real-time OS analysis control
RtosResource
Plug-in for real-time OS resource information display
SampleProjects
Sample project folder for CS+

Figure 1-10. Folder Configuration (Source Release Version) [RI850V4 V1.*xx.xx*]




v850e_ghs
Project file for Green Hills Compiler
r22
22-register mode
r26
26-register mode
r32
32-register mode
v850e2m_cx
Project file for CX [V850E2M]
r22
22-register mode
r26
26-register mode
r32
32-register mode
v850e2m_ghs
Project file for Green Hills Compiler [V850E2M]
r22
22-register mode
r26
26-register mode
r32
32-register mode
kernel
Folder containing the kernel source
AZ850V4
Program analyzer relation
bin
Executable file of program analyzer
hlp
Online help file for program analyzer
lib850
Object files of AZ monitor
r22
22-register mode
r26
26-register mode
r32
32-register mode
smp850
Sample program of user-own coding module for AZ monitor
Plugins
Plug-in relation for CS+
RtosBuild
Plug-in for real-time OS build setting
RtosControl
Plug-in for real-time OS analysis control
RtosResource
Plug-in for real-time OS resource information display
SampleProjects
Sample project folder for CS+

1.4.6 RI850MP

The following shows the folder configuration when the files stored in the RI850MP distribution media have been installed.

Figure 1-11. Folder Configuration (Object Release Version) [RI850MP]

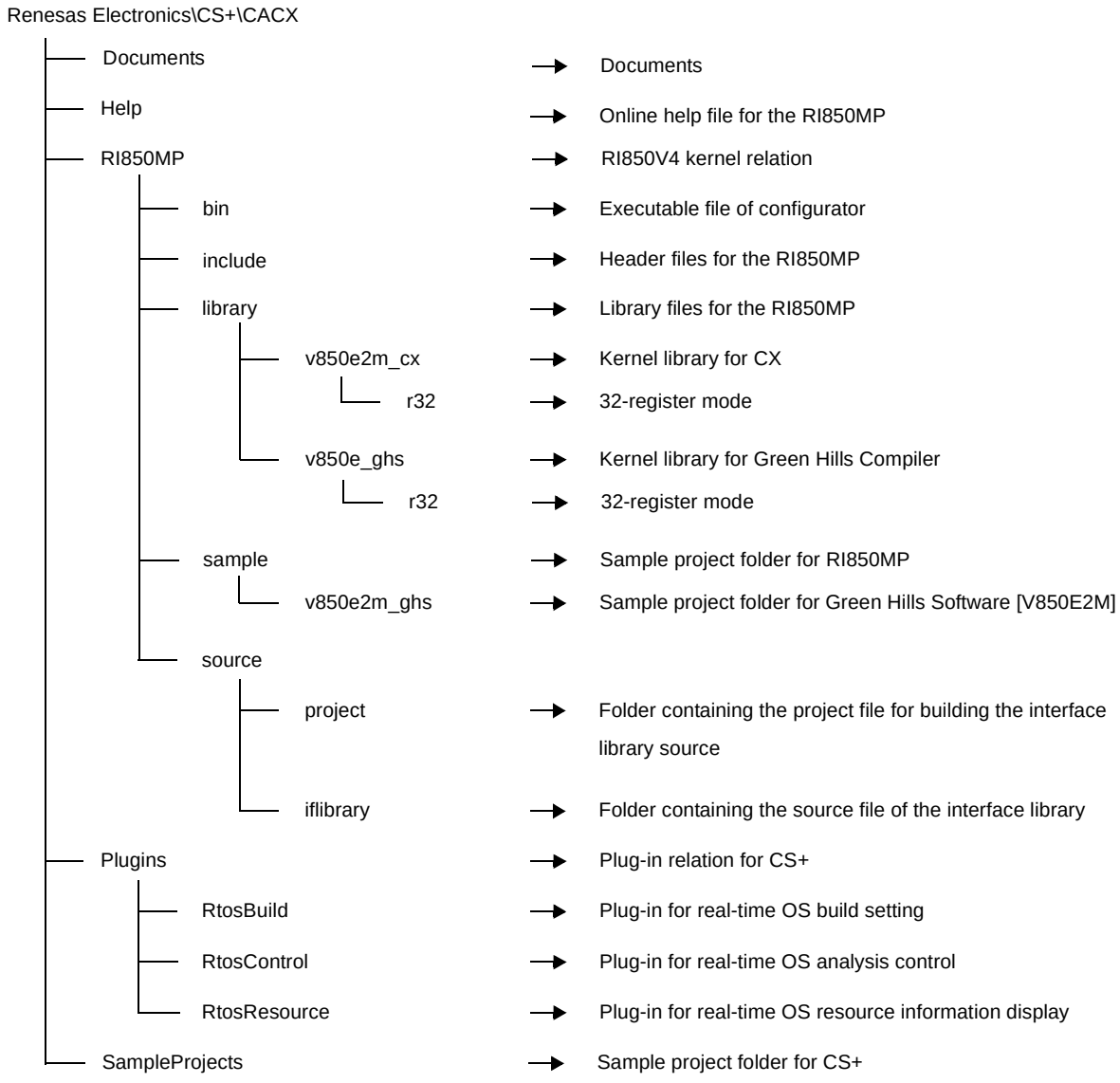
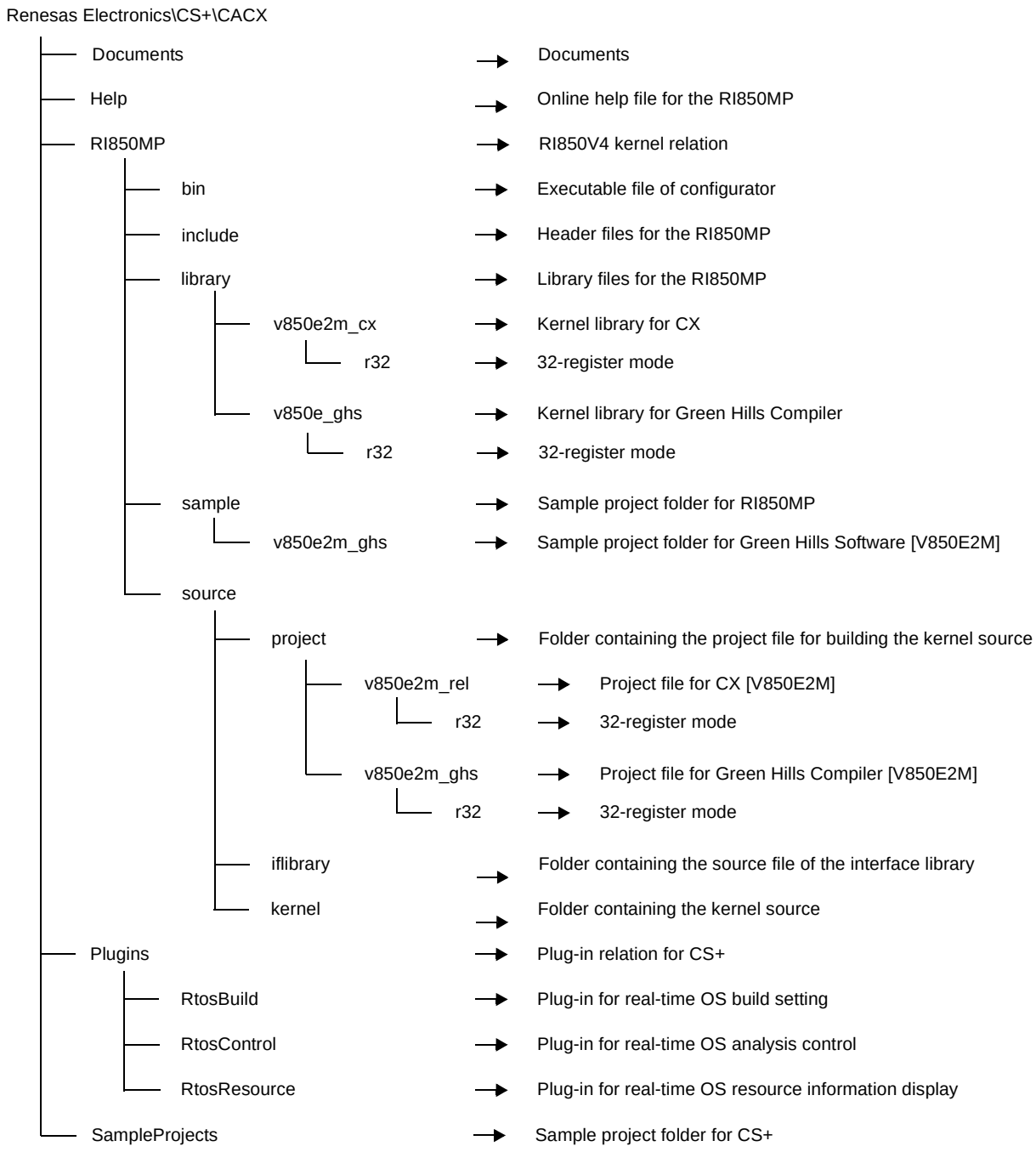


Figure 1-12. Folder Configuration (Source Release Version) [RI850MP]


Renesas Electronics\CS+\CACX
Documents → Documents
Help → Online help file for the RI850MP
RI850MP → RI850V4 kernel relation
bin → Executable file of configurator
include → Header files for the RI850MP
library → Library files for the RI850MP
v850e2m_cx → Kernel library for CX
r32 → 32-register mode
v850e_ghs → Kernel library for Green Hills Compiler
r32 → 32-register mode
sample → Sample project folder for RI850MP
v850e2m_ghs → Sample project folder for Green Hills Software [V850E2M]
source
project → Folder containing the project file for building the kernel source
v850e2m_rel → Project file for CX [V850E2M]
r32 → 32-register mode
v850e2m_ghs → Project file for Green Hills Compiler [V850E2M]
r32 → 32-register mode
iflibrary → Folder containing the source file of the interface library
kernel → Folder containing the kernel source
Plugins → Plug-in relation for CS+
RtosBuild → Plug-in for real-time OS build setting
RtosControl → Plug-in for real-time OS analysis control
RtosResource → Plug-in for real-time OS resource information display
SampleProjects → Sample project folder for CS+

1.4.7 RI78V4 V1.*xx.xx*

The following shows the folder configuration when the files stored in the RI78V4 V1.*xx.xx* distribution media have been installed.

Figure 1-13. Folder Configuration (Object Release Version) [RI78V4 V1.*xx.xx*]

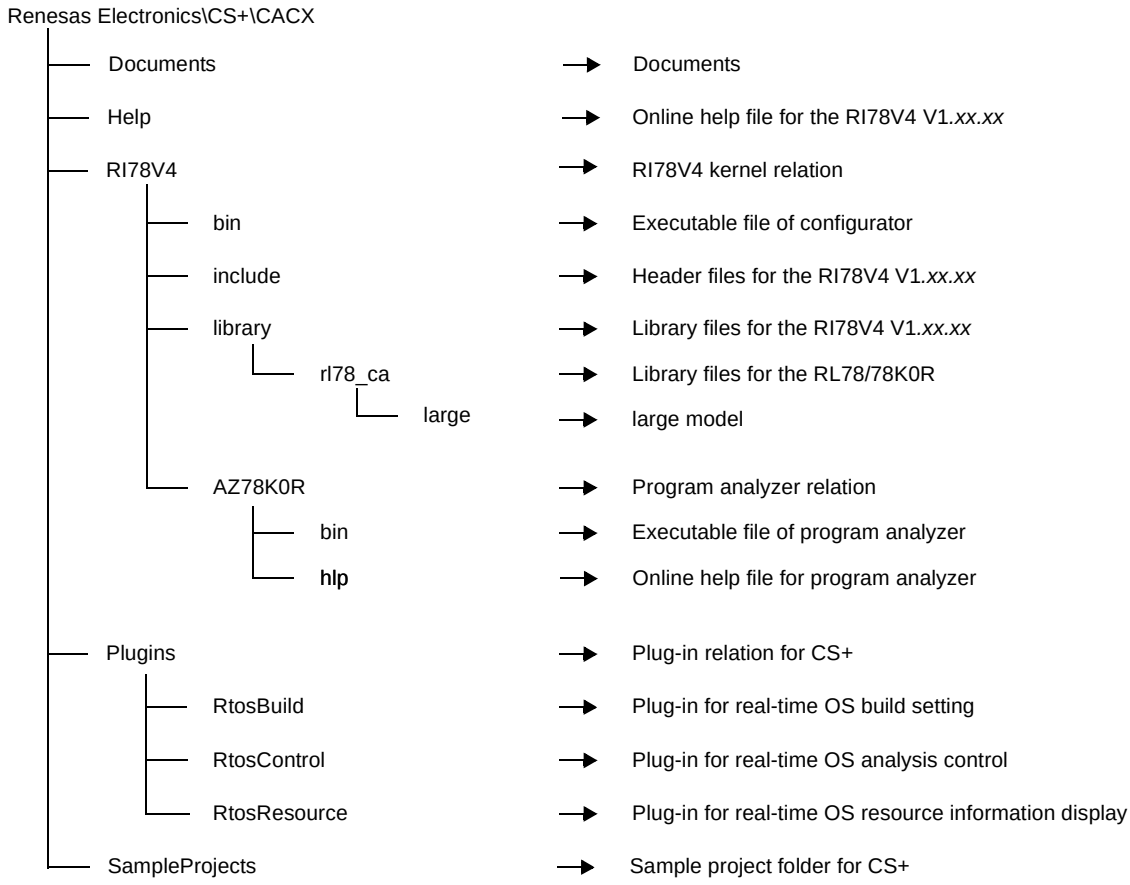
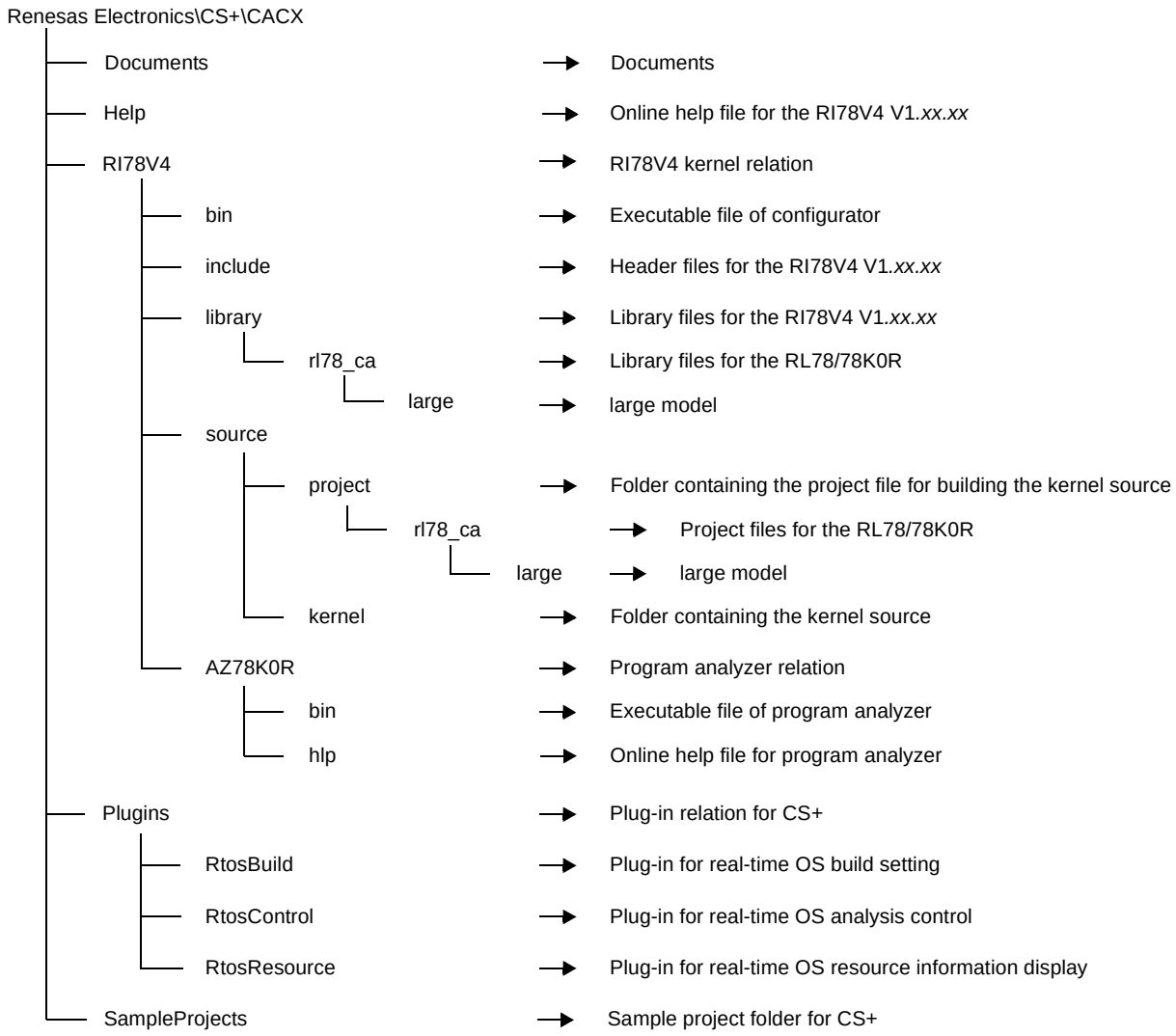


Figure 1-14. Folder Configuration (Source Release Version) [RI78V4 V1.*xx.xx*]


Renesas Electronics\CS+\CACX
Documents → Documents
Help → Online help file for the RI78V4 V1.xx.xx
RI78V4 → RI78V4 kernel relation
bin → Executable file of configurator
include → Header files for the RI78V4 V1.xx.xx
library → Library files for the RI78V4 V1.xx.xx
rl78_ca → Library files for the RL78/78K0R
large → large model
source
project → Folder containing the project file for building the kernel source
rl78_ca → Project files for the RL78/78K0R
large → large model
kernel → Folder containing the kernel source
AZ78K0R → Program analyzer relation
bin → Executable file of program analyzer
hlp → Online help file for program analyzer
Plugins → Plug-in relation for CS+
RtosBuild → Plug-in for real-time OS build setting
RtosControl → Plug-in for real-time OS analysis control
RtosResource → Plug-in for real-time OS resource information display
SampleProjects → Sample project folder for CS+

1.5 Execution Environment

The following shows hardware required for the RI series to perform processing.

(1) CPU

The following shows CPU required for the RI series to perform processing.

- RL78 family [RI78V4 V2.*xx.xx*] [RI78V4 V1.*xx.xx*]
- RH850 family (G3K/G3M core) [RI850V4 V2.*xx.xx*]
- RX family (RX100/RX200 core) [RI600V4]
- RX family (RX600 core) [RI600V4][RI600PX]
- V850 family (V850ES/V850E1/V850E2/V850E2M core) [RI850V4 V1.*xx.xx*]
- V850 family equipped with two V850E2M cores [RI850MP]
- 78K0R microcontrollers [RI78V4 V1.*xx.xx*]

(2) Peripheral controller

To support various execution environments, the RI series extracts hardware-dependent processing as user-own coding module and target-dependent modules, provides it as sample source files. Because the execution environment is supported just by rewriting the user-own coding module and target-dependent modules according to the environment, special peripheral controllers are not required.

Controllers such as a clock controller are required to use the time management functions provided by the RI series, or controllers such as an interrupt controller are required to use the interrupt management functions.

(3) Memory capacity

The following shows the memory capacity required for the RI series to perform processing.

Regarding the figures listed below, the required memory capacity can be minimized by setting limits on the total number of definitions of OS resource-related information defined during configuration and the types of service calls that are used by the system.

- RI78V4 V2.*xx.xx*
  - ROM area: 3 KB or more
  - RAM area: 1 KB or more
- RI850V4 V2.*xx.xx*
  - ROM area: 5.5 KB or more
  - RAM area: 1.1 KB or more
- RI600V4
  - ROM area: 6 KB or more
  - RAM area: 1 KB or more
- RI600PX
  - ROM area: 7 KB or more
  - RAM area: 1 KB or more
- RI850V4 V1.*xx.xx*
  - ROM area: 6 KB or more
  - RAM area: 1 KB or more
- RI850MP
  - ROM area: 6 KB or more
  - RAM area: 1 KB or more
- RI78V4 V1.*xx.xx*
  - ROM area: 1 KB or more
  - RAM area: 1 KB or more

(4) Supported debug tool

Below is described the environment necessary for running the resource information tool and program analyzer tool/ task analyzer tool.

Table 1-1. Supported Resource Information Display Tool

Debug Tools	RI78V4 V2.*xx.xx*	RI850V4 V2.*xx.xx*	RI600V4	RI600PX	RI850V4 V1.*xx.xx*	RI850MP	RI78V4 V1.*xx.xx*
IECUBE	OK	NG	NG	NG	OK	NG	OK
IECUBE2	NG	NG	NG	NG	OK	NG	NG
MINICUBE	NG	NG	NG	NG	OK	OK	NG
MINICUBE2	NG	NG	NG	NG	OK	NG	OK
E1	OK	OK	OK	OK	OK	OK	OK
E20	OK	OK	OK	OK	OK	OK	OK
EZ Emulator	NG	NG	OK	NG	OK	NG	OK
Simulator	OK	OK	OK	OK	OK	NG	OK

OK: It can be used by this combination.
NG: It can't be used by this combination.

Table 1-2. Supported Program Analyze Tool

	RI850V4 V1.xx.xx	RI78V4 V1.*xx.xx*
IECUBE	OK	OK[Note 1]
IECUBE2	OK	NG
MINICUBE	OK[Note 2]	NG
MINICUBE2	OK[Note 2]	NG
E1	OK[Note 2]	NG
E20	OK[Note 2]	NG
EZ Emulator	OK[Note 2]	NG
Simulator	OK	OK[Note 1]

OK: It can be used by this combination.
NG: It can't be used by this combination.

Notes 1. It can be used only by a hardware trace mode. It can't be used by a software trace mode.
2. It can be used only by a software trace mode. It can't be used by a hardware trace mode.

Table 1-3. Supported Task Analyzer Too

	RI78V4 V2.*xx.xx*	RI850V4 V2.*xx.xx*	RI600V4
IECUBE	OK	NG	NG
IECUBE2	NG	NG	NG
MINICUBE	NG	NG	NG
MINICUBE2	NG	NG	NG
E1	OK[Note 1]	OK[Note 2]	OK[Note 2]
E20	OK[Note 1]	OK	OK
EZ Emulator	NG	NG	OK[Note 1]
Simulator	OK	OK	OK

OK: It can be used by this combination.

NG: It can't be used by this combination.

Notes 1. It can be used only by a software trace mode. It can't be used by a hardware trace mode.

2. The acquisition of trace information is limited to a maximum of 256 cycles in hardware trace mode.

CHAPTER 2 FUNCTIONS

This chapter describes how to install the real-time OS package, and how to start tools provided by it.

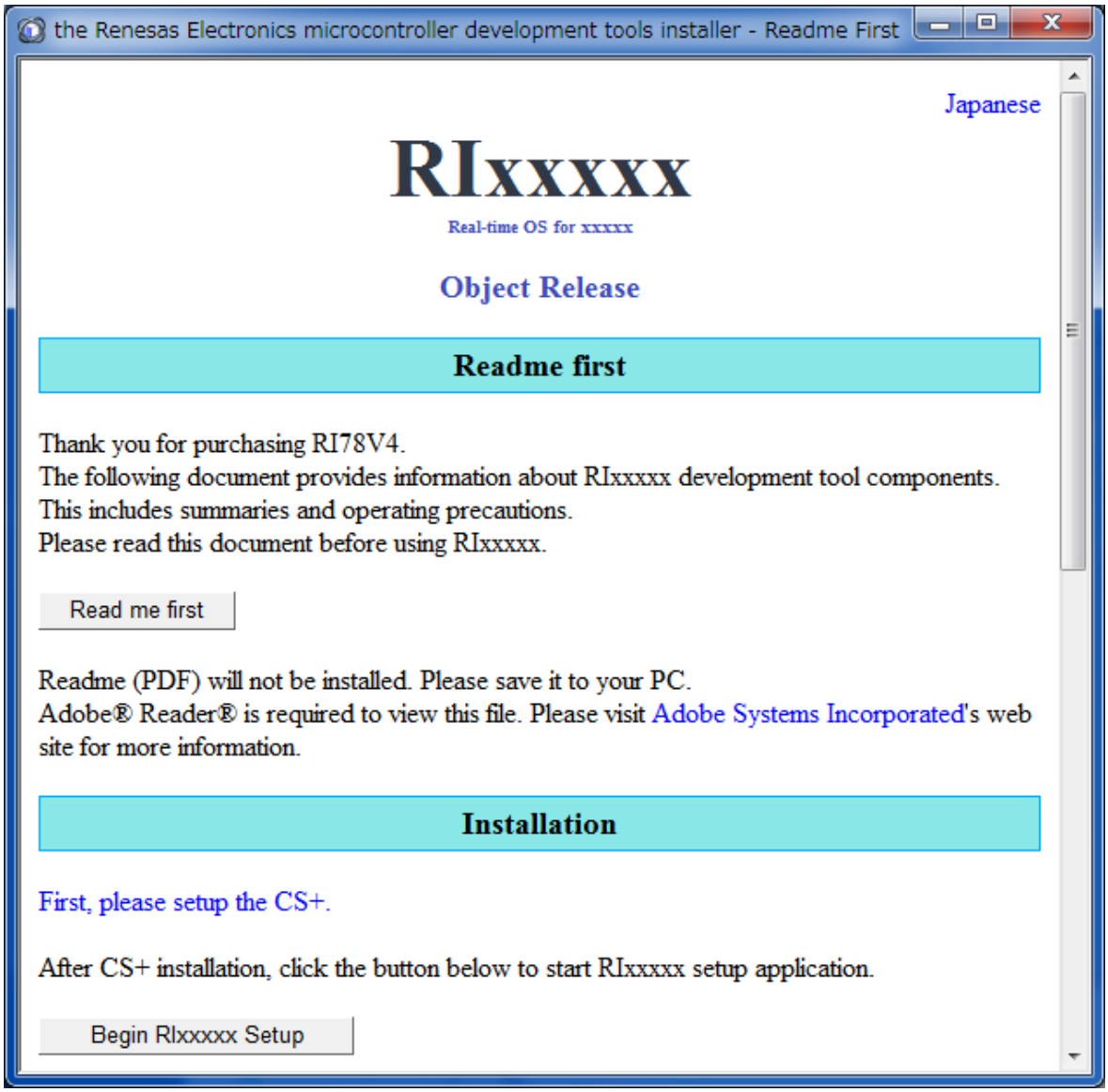
2.1 Installing Real-time OS Package

This section describes how to install real-time OS package.

(1) Insert the CD-ROM into the drive.

The Preparing to Install page appears automatically.
After, please operate with displayed contents.

Figure 2-1. Preparing to Install Page


the Renesas Electronics microcontroller development tools installer - Readme First
Japanese
RIxxxxx
Real-time OS for xxxxx
Object Release
Readme first
Thank you for purchasing RI78V4.
The following document provides information about RIxxxxx development tool components.
This includes summaries and operating precautions.
Please read this document before using RIxxxxx.
Read me first
Readme (PDF) will not be installed. Please save it to your PC.
Adobe® Reader® is required to view this file. Please visit Adobe Systems Incorporated's web site for more information.
Installation
First, please setup the CS+.
After CS+ installation, click the button below to start RIxxxxx setup application.
Begin RIxxxxx Setup

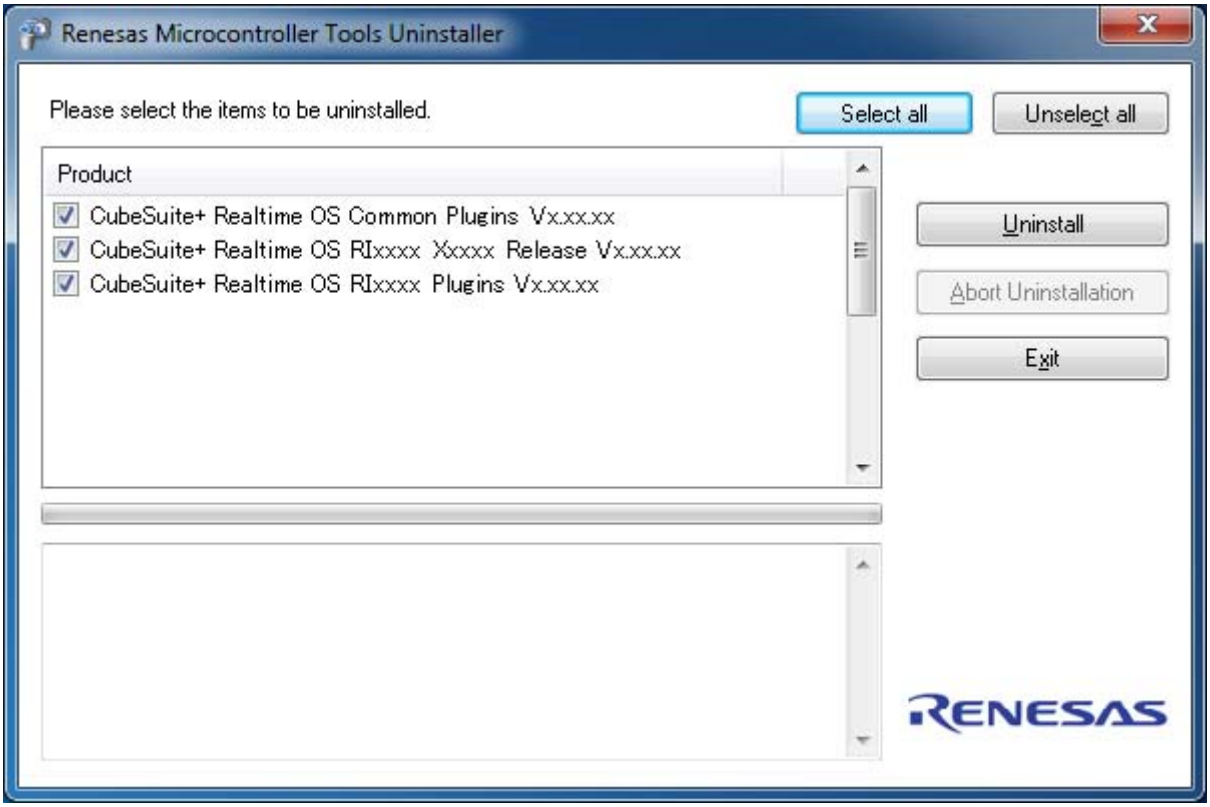
Remark If the page does not appear automatically, open "Install.hta" in the CD-ROM.

2.2 Uninstalling Real-time OS Package

You can uninstall real-time OS package using the CS+ Uninstaller.

From the Windows [Start] menu, select [Programs] >> [Renesas Electronics CS+] >> [CS+ Uninstaller]. The uninstallation begins.

Figure 2-2. CS+ Uninstaller Window



The CS+ Uninstaller window looks up all CS+ products and real-time OS package products you have installed, and displays them in a list of check boxes.

Click [Select all] to select all the check boxes.

Click [Unselect all] to clear all the check boxes.

Click [Uninstall] to uninstall the selected products.

Caution If all products are uninstalled, the CS+ Uninstaller will be also uninstalled automatically.

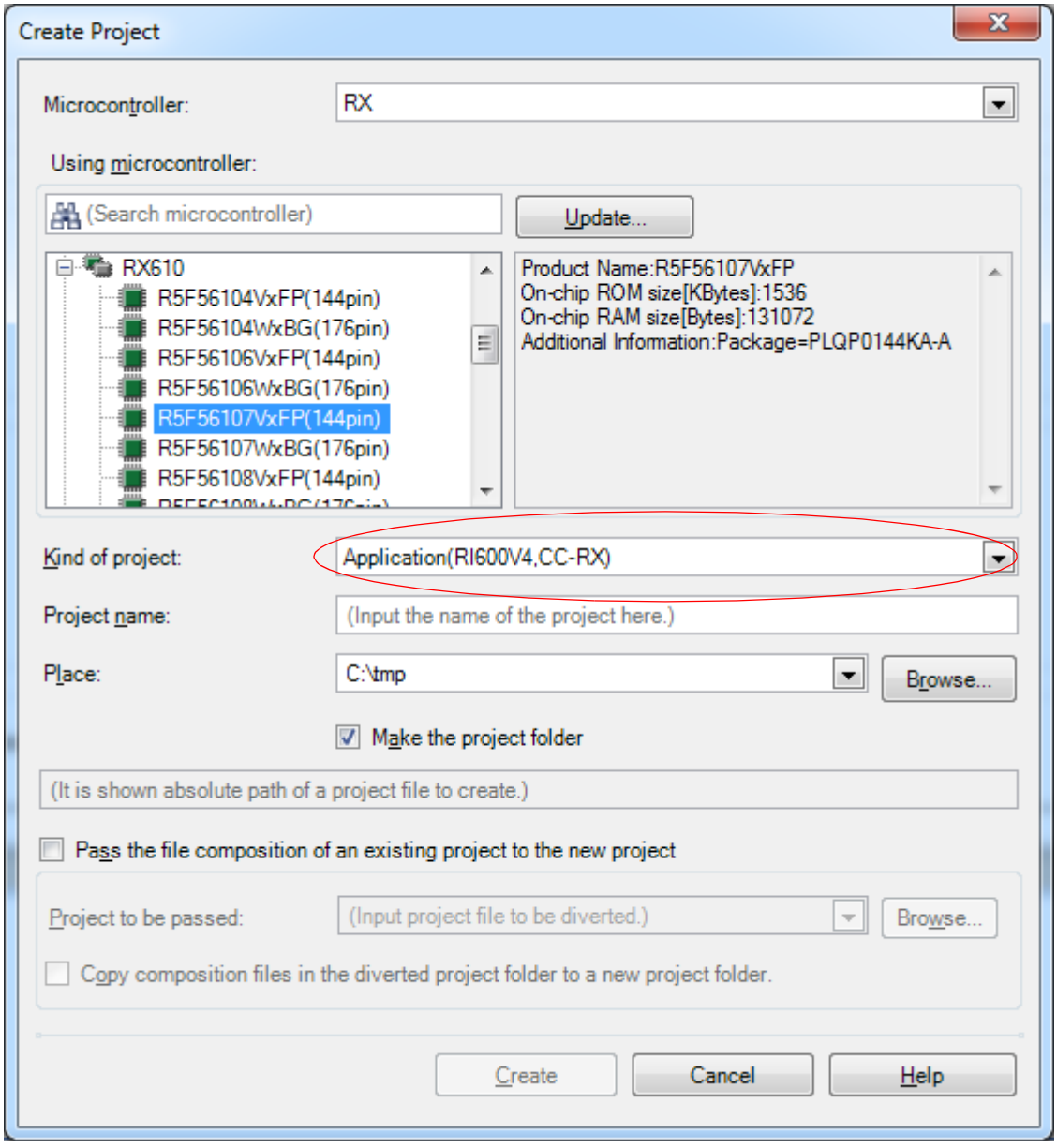
Remark See "CS+ Integrated Development Environment User's Manual: Start" for details.

2.3 Create a Project for RI Series

The creation of a project for RI series is performed with the Create Project dialog box (See the CS+ Integrated Development Environment User's Manual: Start "2.6.2 Create a new project").

From the [Project] menu, select [Create New Project...], the following dialog box will open.

Figure 2-3. Create Project Dialog Box



Select the item below on [Kind of project].

- Application(*RI series name*, *Build tool name*)
  Select this to generate a project for RI series.

2.4 Convert a CubeSuite Project into a CS+ Project

In the RI series, you can convert a CubeSuite project into a CS+ project automatically by selecting [Open Project...] from the [Project] menu with CS+.

- The real-time OS name in use will be changed at the same time (RX78K0R -> RI78V4 V1.*xx.xx*, RX850V4 -> RI850V4 V1.*xx.xx*)
- With the change of the real-time OS name, substitution processing will be performed for the following: system configuration file that was used in a CubeSuite project; real-time OS name described in the link directive file; version; and memory area
- Since substitution processing will be performed on your system configuration file and link directive file, backup copies of the original files will first be made. Name the copied file as original-file-name.(dot)old-real-time OS-name.
- Because substitution processing will be performed, do not add a write inhibit attribute to any file or folder to be converted. If it has a write inhibit attribute, no substitution processing will be performed.
- The character code of the file generated after substitution will be the same as the character code before substitution. If the character code cannot be identified, the Windows default character code will be used.
- After processing of conversion to CS+, output (rebuild) again the file generated from the system configuration file.

Caution It cannot convert a project directly “RX78K0R -> RI78V4 V2.*xx.xx*” or “RX850V4 -> RI850V4 V2.*xx.xx”.*In this case, first, converts project “RX78K0R -> RI78V4 V1.*xx.xx*” or “RX850V4 -> RI850V4 V1.*xx.xx”,* next, uses project diversion function.

2.5 Convert a HEW Project into a CS+ Project

When a HEW (hereafter abbreviated "HEW") project is loaded, if the real-time OS was in use in that project, then it will be reflected in the properties of the build tool.

The real-time OS settings in the HEW project are shown below.

- On the RX Standard Toolchain [RI600/4] tab, in the [Configuration] category, the [Miscellaneous options] settings except for the "[-v] Displays the command option descriptions and detailed information on the version" option.
- On the RX Standard Toolchain [RI600/4] tab, in the [Configuration] category, the [User-defined options] settings.
- On the RX Standard Toolchain [RI600/4] tab, in the [Mkritbl] category, the "Search directory of MRC files" settings

2.6 Convert a Project “RI78V4 V1.xx.xx -> V2.xx.xx“ or “RI850V4 V1.xx.xx -> V2.xx.xx“

When converting a project of “RI78V4 V1.xx.xx“ or “RI850V4 V1.xx.xx” to a project of “RI78V4 V2.xx.xx” or “RI850V4 V2.xx.xx”, you convert using “the project diversion function”. In other words, you make a project of “RI78V4 V1.xx.xx” or “RI850V4 V1.xx.xx” the diversion subject , choose “RI78V4 V2.xx.xx” or “RI850V4 V2.xx.xx” as the kind of made projects and make a project. In detail, please refer to the manual “CS+ Project Operation“.

2.7 Start Resource Information Tool

When a program using the real-time OS feature is downloaded, the Realtime OS Resource Information panel is opened automatically.

From [View] menu, select [Realtime OS] >> [Resource Information]. The Realtime OS Resource Information panel opens.

Figure 2-4. [Resource Information] Item

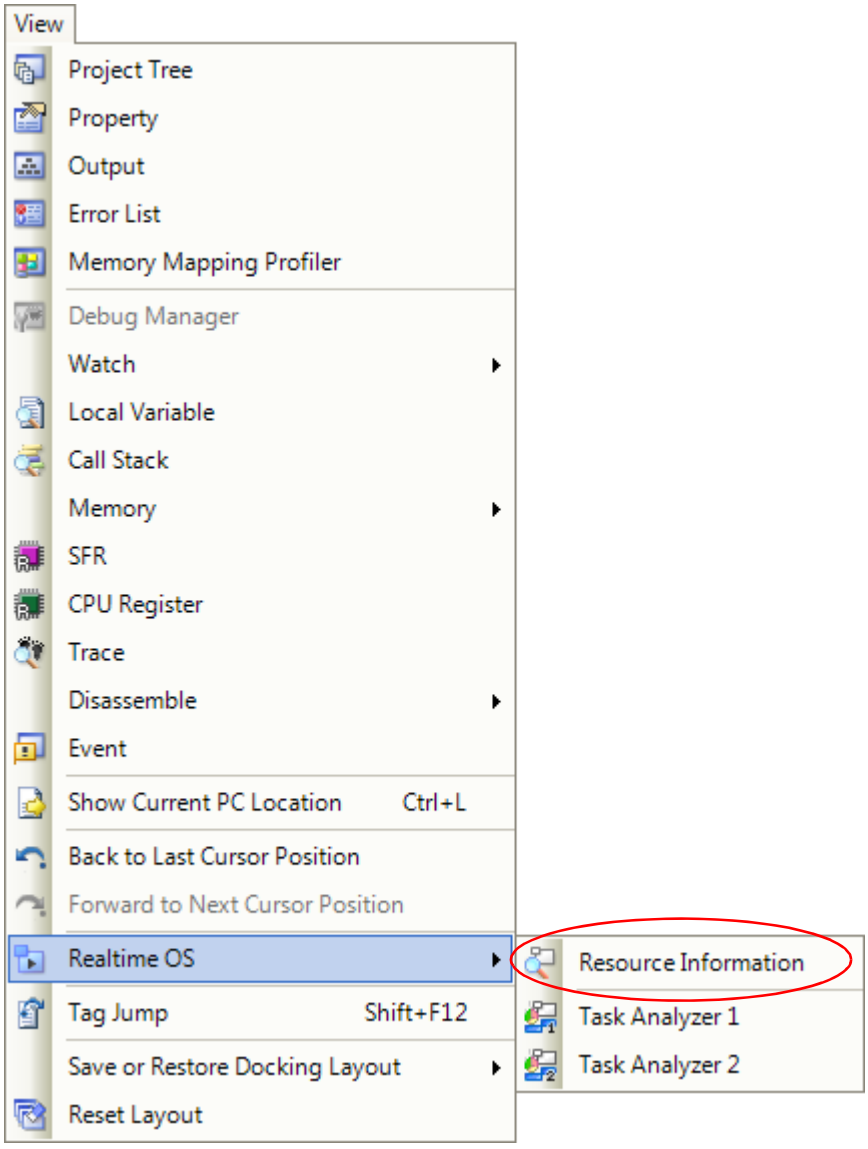
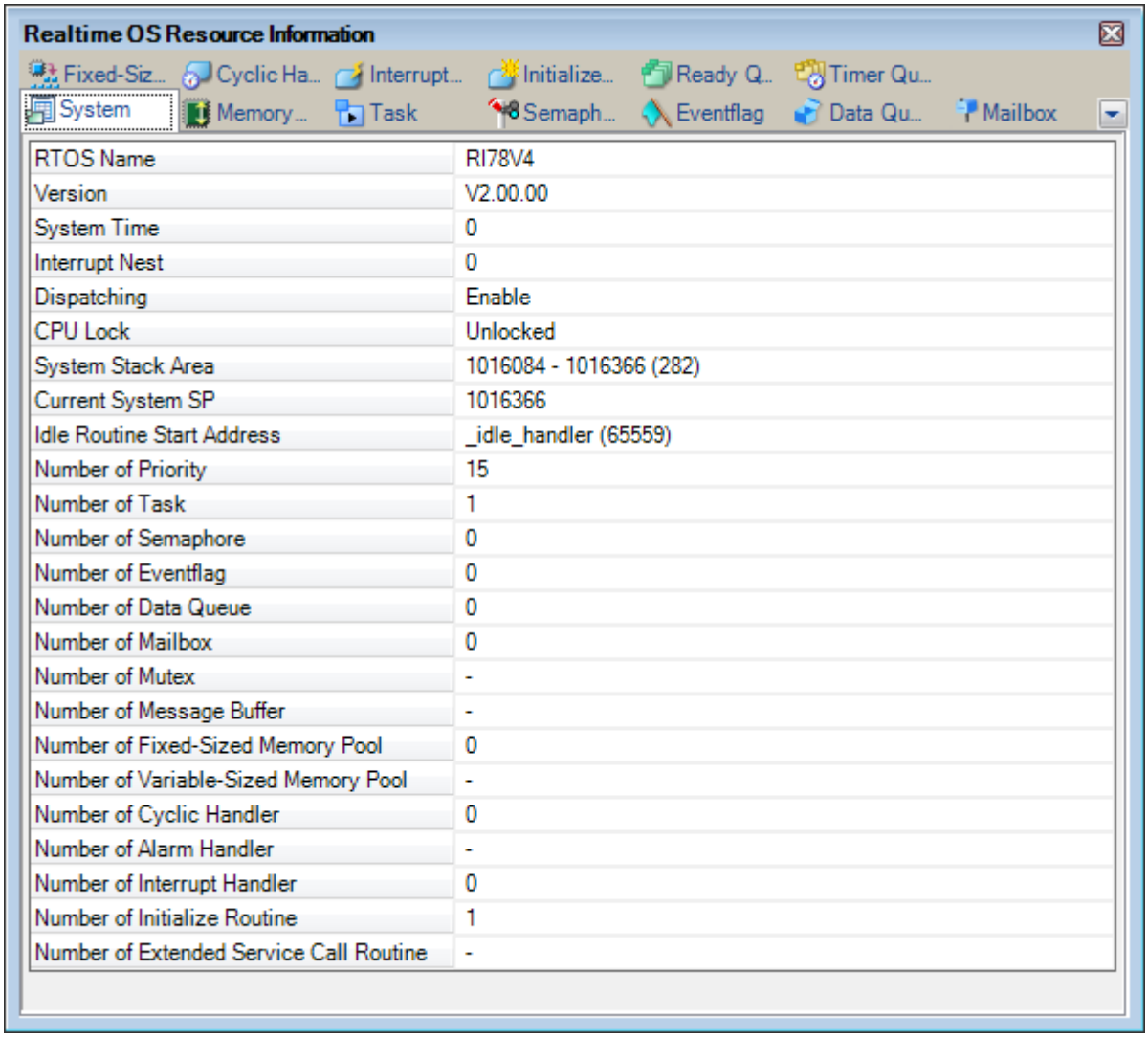


Figure 2-5. Realtime OS Resource Information Panel



RTOS Name	RI78V4
Version	V2.00.00
System Time	0
Interrupt Nest	0
Dispatching	Enable
CPU Lock	Unlocked
System Stack Area	1016084 - 1016366 (282)
Current System SP	1016366
Idle Routine Start Address	_idle_handler (65559)
Number of Priority	15
Number of Task	1
Number of Semaphore	0
Number of Eventflag	0
Number of Data Queue	0
Number of Mailbox	0
Number of Mutex	-
Number of Message Buffer	-
Number of Fixed-Sized Memory Pool	0
Number of Variable-Sized Memory Pool	-
Number of Cyclic Handler	0
Number of Alarm Handler	-
Number of Interrupt Handler	0
Number of Initialize Routine	1
Number of Extended Service Call Routine	-

Remarks 1. When resource information tool has started, it isn't possible to start more resource information tool.

2. See "RI78V4 V2.*xx*.*xx*/RI850V4 V2.*xx*.*xx*/RI600V4/RI850V4/RI850MP/RI78V4 Real-Time Operating System User's Manual: Debug" for details.

2.8 Start Task Analyzer Tool [RI78V4 V2.xx.xx][RI850V4 V2.xx.xx][RI600V4]

From [View] menu, select [Realtime OS] >> [Task Analyzer 1] or [Task Analyzer 2]. The Realtime OS Task Analyzer panel opens.

Figure 2-6. [Task Analyzer] Item

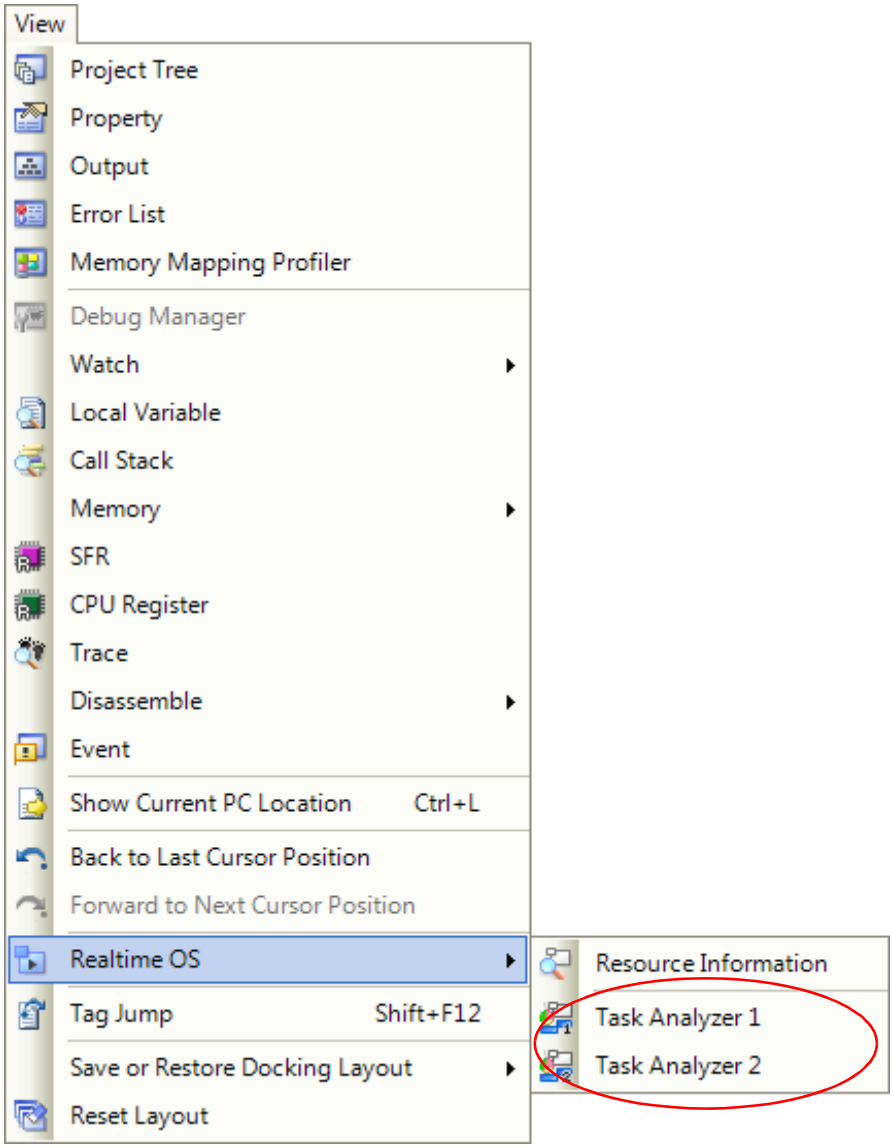
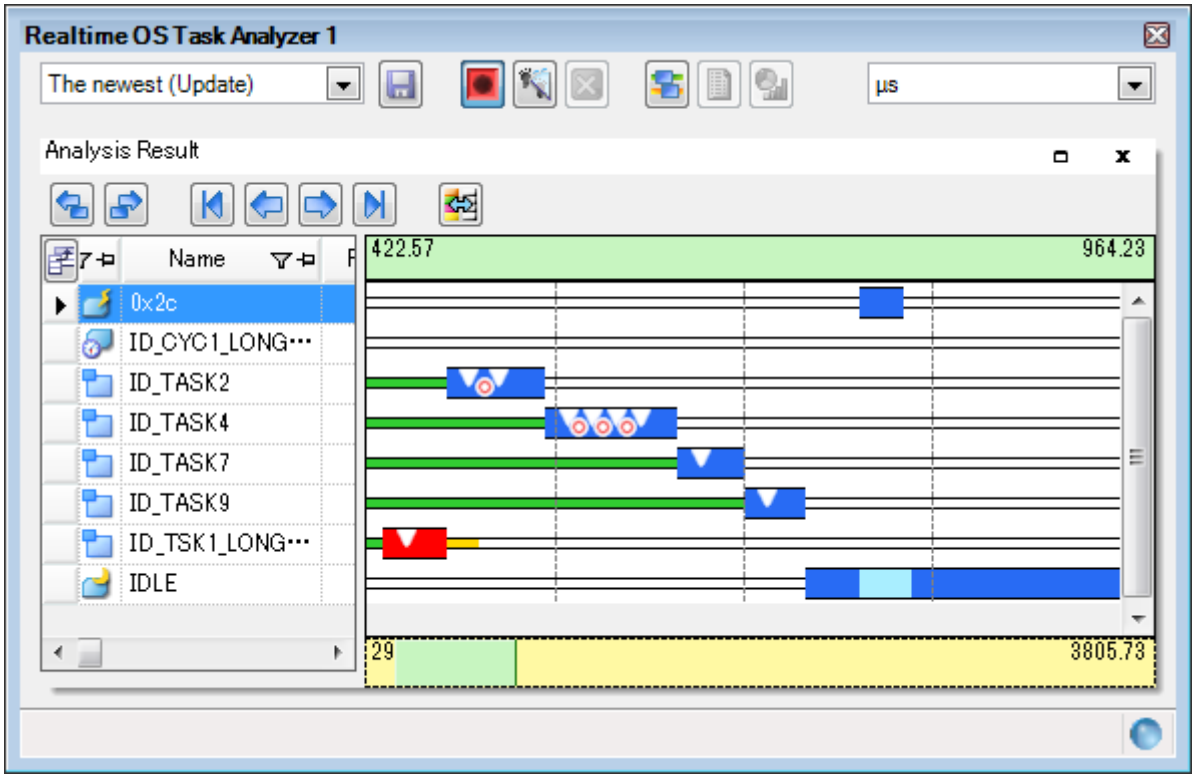


Figure 2-7. Realtime OS Task Analyzer Panel



Remarks 1. Up to two panels can be opened by the task analyzer tool: the Realtime OS Task Analyzer1 panel and the Realtime OS Task Analyzer2 panel.

2. See "RI78V4 V2.*xx.xx*/RI850V4 V2.*xx.xx*/RI600V4 Real-Time Operating System User's Manual: Analysis" for details.

2.9 Start Program Analyzer Tool [RI850V4 V1.xx.xx][RI78V4 V1.xx.xx]

From [View] menu, select [Realtime OS] >> [Program Analyzer], program analyzer is started.
From [View] menu, select [Realtime OS] >> [Program Analyzer]. The program analyzer is started.

Figure 2-8. [Program Analyzer] Item

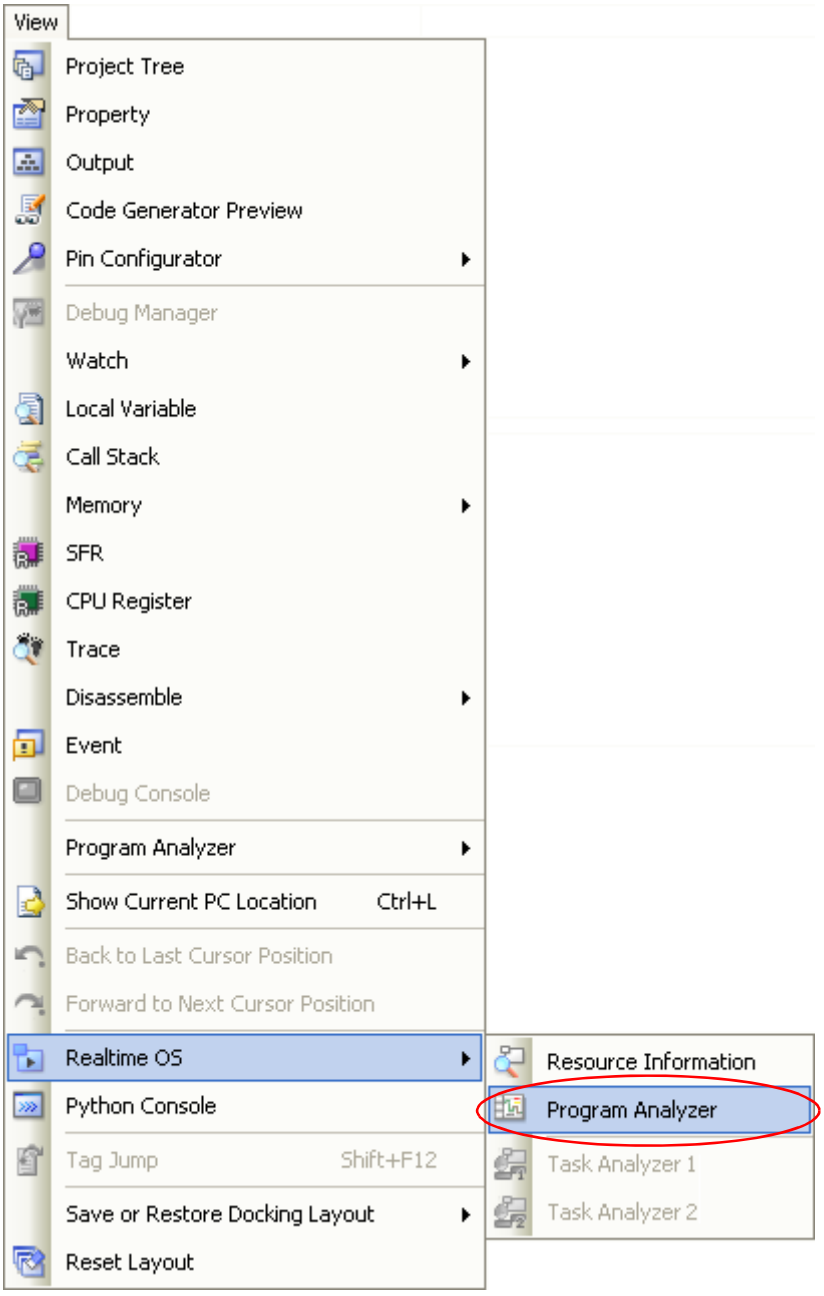
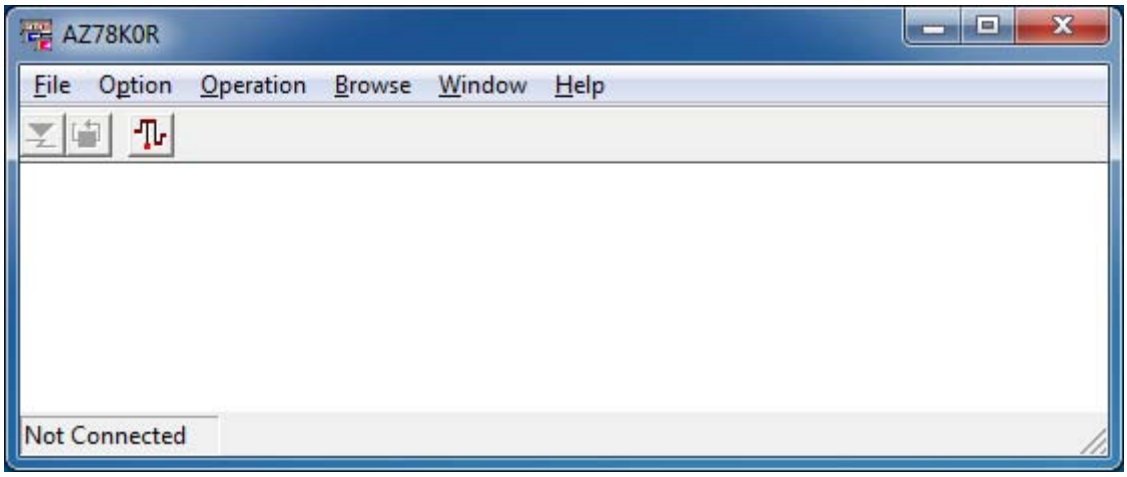


Figure 2-9. Program Analyzer Window



Remarks 1. When program analyzer has started, it isn't possible to start more program analyzer.

2. See "RI850V4/RI78V4 Real-Time Operating System User's Manual: Analysis" for details.

APPENDIX A WINDOW REFERENCE

This appendix describes the windows related to tools offered in the real-time OS package.

A.1 Description

Below is a list of the windows related to tools offered in the real-time OS package.

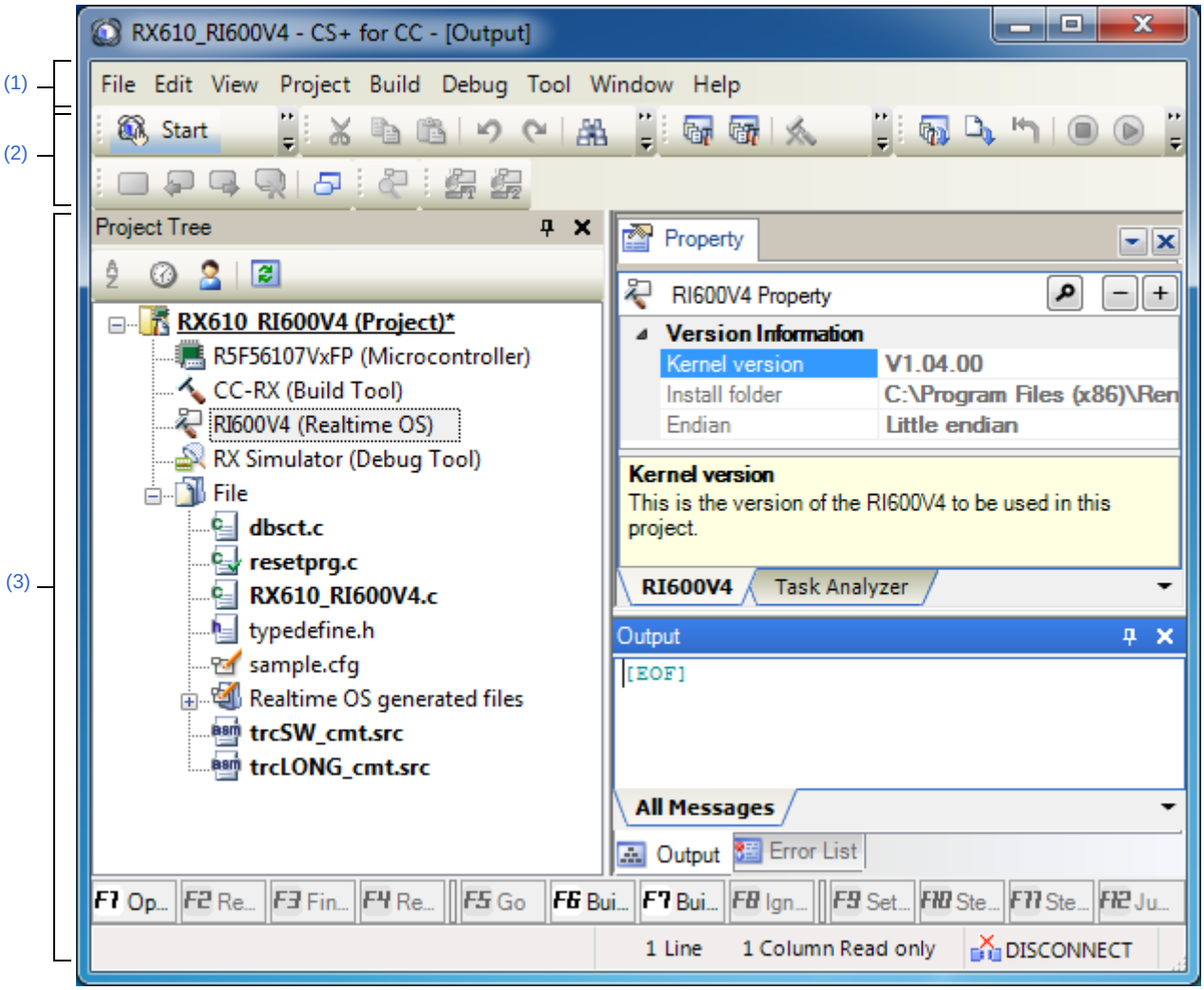
Table A-1. Window/Panel/Dialog Box List

Window/Panel/Dialog Box Name	Function Description
Main window	The first window opened when CS+ starts.
Create Project dialog box	Creates a new project or subproject. (See "CS+ Integrated Development Environment User's Manual: Start" for details.)
Project Tree panel	Displays the project elements as a tree. (See "RI78V4 V2.*xx.xx*/RI850V4 V2.*xx.xx*/RI600V4/RI600PX/RI850V4/RI850MP/RI78V4 Real-Time Operating System User's Manual: Coding" for details.)
Property panel	Displays details and change the settings of the node selected in the Project Tree panel. (See "RI78V4 V2.*xx.xx*/RI850V4 V2.*xx.xx*/RI600V4/RI600PX/RI850V4/RI850MP/RI78V4 Real-Time Operating System User's Manual: Coding" for details.)
Realtime OS Resource Information panel	Displays real-time OS resource information (system information, memory area information, etc.). (See "RI78V4 V2.*xx.xx*/RI850V4 V2.*xx.xx*/RI600V4/RI600PX/RI850V4/RI850MP/RI78V4 Real-Time Operating System User's Manual: Debug" for details.)
Realtime OS Task Analyzer panel [RI78V4 V2.*xx.xx*][RI850V4 V2.*xx.xx*] [RI600V4]	Displays information required to analyze the processing program (the status of execution transition of programs being processed, usage of real-time OS resources, usage of the CPU, etc.). (See "RI78V4 V2.*xx.xx*/RI850V4 V2.*xx.xx*/RI600V4 Real-Time Operating System User's Manual: Analysis" for details.)
Program analyzer window (AZ850V4 window) [RI850V4 V1.*xx.xx*]	Executes basic operation of program analyzer. (See "RI850V4 Real-Time Operating System User's Manual: Analysis" for details.)
Program analyzer window (AZ78K0R window) [RI78V4 V1.*xx.xx*]	Executes basic operation of program analyzer. (See "RI78V4 Real-Time Operating System User's Manual: Analysis" for details.)

Main window

This is the start-up window that opens when CS+ is launched.
In this window, you can control the user program execution and open panels.

Figure A-1. Main Window



The following items are explained here.

- [How to open]
- [Description of each area]

[How to open]

- Select Windows [Start] >> [Programs] >> [Renesas Electronics CS+] >> [CS+].

[Description of each area]

(1) Menu bar

This displays menus about real-time OS package.

(a) [View]

Realtime OS	Displays a cascading menu for relating to real-time OS package.
Resource Information	Opens the Realtime OS Resource Information panel.
Program Analyzer	Program analyzer is started. [RI850V4 V1.*xx.xx*] Opens the AZ850V4 window. [RI78V4 V1.xx.xx] Opens the AZ78K0R window.
Task Analyzer 1	Opens the Realtime OS Task Analyzer 1 panel. [RI78V4 V2.*xx.xx*][RI850V4 V2.*xx.xx*][RI600V4]
Task Analyzer 2	Opens the Realtime OS Task Analyzer 2 panel. [RI78V4 V2.*xx.xx*][RI850V4 V2.*xx.xx*][RI600V4]

(2) Toolbar

The toolbar shows command buttons relating to real-time OS package.

	Opens the Realtime OS Resource Information panel.
	Opens the Realtime OS Task Analyzer 1 panel. [RI78V4 V2.*xx.xx*] [RI850V4 V2.*xx.xx*][RI600V4]
	Opens the Realtime OS Task Analyzer 2 panel. [RI78V4 V2.*xx.xx*] [RI850V4 V2.*xx.xx*][RI600V4]

(3) Panel display area

Panels are displayed in this area.

Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	Oct 01, 2011	-	First Edition issued
1.01	Apr 01, 2012	7, 8, 10, 16, 17, 26, 27, 41	Added the following: RI600PX
		7, 19, 21, 24, 25, 27, 39, 40, 41, 43	Changed as follows: Performance analyzer -> Program analyzer
		9	1.3.1 RI600V4 Added the following: "(2)User-own coding module" - "System initializing process" Section information
1.02	Sep 01, 2012	7	1.2 Features Added the following: "(4)Support tools related to real-time OS" Task analyzer tool [RI600V4]
		14, 15	1.4.1 RI600V4 Added the following: "Figure 1-1. Folder Configuration (Object Release Version) [RI600V4]", "Figure 1-2. Folder Configuration (Source Release Version) [RI600V4]" Plug-in for task analyzer tool
		27	1.5 Execution Environment Added the following: "Table 1-1. Supported Debug Tool" Task Analyzer Tool
		39	Added the following: 2.7 Start Task Analyzer Tool [RI600V4]
		43	A.1 Description Added the following: "Table A-1. Window/Panel/Dialog Box List" Task analyzer panel [RI600V4]
		45	Main window Added the following: "[Description of each area]" - "(1)Menu bar", "(2)Toolbar" Description of the task analyzer tool
1.03	Mar 03, 2014	-	Added the “RI850V4 V2.*xx.xx*“

Rev.	Date	Description	
		Page	Summary
1.04	Mar 25, 2015	-	Added the “RI78V4 V2.*xx.xx*“
		-	Changed the “CubeSuite+“ to “CS+“
		-	Changed the image for CS+
1.05	Sep 30, 2015	18	Added the folder for Green Hills Software to “Folder Configuration (Object Release Version [RI850V4 V2.*xx.xx*]“
		19	Added the folder for Green Hills Software to “Folder Configuration (Source Release Version [RI850V4 V2.*xx.xx*]“
		30	Changed the description as following - RH850 Family ->- - RH850 Family (G3K/G3M core)
		30	Changed the description as following - RX family (RX200 core) [RI600V4] ->- - RX family (RX100/RX200 core) [RI600V4]

RI Series User's Manual:
Start

Publication Date: Rev.1.00 Oct 01, 2011
Rev.1.05 Sep 30, 2015

Published by: Renesas Electronics Corporation

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

California Eastern Laboratories, Inc.
4590 Patrick Henry Drive, Santa Clara, California 95054-1817, U.S.A.
Tel: +1-408-919-2500, Fax: +1-408-988-0279

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HALII Stage, Indiranagar, Bangalore, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

© 2015 Renesas Electronics Corporation. All rights reserved.

Colophon 4.0

RI Series

Renesas Electronics Corporation

R20UT0751EJ0105